In [3]:

```
!pip install torchsummary
```

```
Collecting torchsummary
  Downloading torchsummary-1.5.1-py3-none-any.whl (2.8 kB)
Installing collected packages: torchsummary
Successfully installed torchsummary-1.5.1
```

In [4]:

```
import numpy as np

import scipy.io
import os
from numpy.linalg import norm,det,inv,svd
from scipy.linalg import rq
import math
import matplotlib.pyplot as plt
import numpy as np
import math
import random
import sys
from scipy import ndimage,spatial
from tqdm.notebook import trange,tqdm
import torch
import torch.nn as nn
import torch.optim as optim
from torch.optim import lr_scheduler
from torch.autograd import Variable
import torchvision
from torchvision import datasets,models,transforms
from  torch.utils.data import Dataset,DataLoader,ConcatDataset
from skimage import io,transform,data
from torchvision import transforms,utils
import  os
import sklearn.svm
import cv2
from os.path import exists
import pandas as pd
import PIL
from sklearn.metrics.cluster import completeness_score
from sklearn.cluster import KMeans
from tqdm import tqdm,tqdm_notebook
from functools import partial
from  torchsummary import summary
from torchvision.datasets import ImageFolder
from torch.utils.data.sampler import SubsetRandomSampler
```

In [5]:

```
class Image:
    def __init__(self,img,position):
        self.img = img
        self.position  = position

inliner_matchset = []
def features_matching(a,keypointlength,threshold):
    bestmatch = np.empty((keypointlength), dtype=np.int16)
    img1index = np.empty((keypointlength),dtype=np.init16)
    distance = np.empty((keypointlength))
    index =0
    for j in range(0,keypointlength):
        x=a[j]
        listx = x.tolist()
        x.sort()
        minval1=x[0]
        minval2=x[1]
```

```
        itemindex1 = listx.index(minval1)
        itemindex2 = listx.index(minval2)
        ratio = minval1/minval2

        if ratio < threshold:
            bestmatch[index] = itemindex1
            distance[index] = minval1
            img1index[index] = j
            index = index + 1
    return [cv2.DMatch(img1index[i],bestmatch[i].astype(int),distance[i]) for i in range
(0,index)]

def compute_Hmography(im1_pts,im2_pts):
    num_matches=len(im1_pts)
    num_rows = 2*num_matches
    num_cols = 9
    A_matrix_shape = (num_rows,num_cols)
    A = np.zeros(A_matrix_shape)
    a_index = 0
    for i in range(0,num_matches):
        (a_x,a_y) = im1_pts[i]
        (b_x,b_y) = im2_pts[i]
        row1 = [a_x,a_y,1,0,0,0,-b_x*a_x,-b_x*a_y,-b_x]
        row2 = [0,0,0,a_x,a_y,1,-b_y*a_x,-b_y*a_y,-b_y]
        A[a_index] = row1

        A[a_index+1] = row2
        a_index += 2

    U,s,Vt = np.linalg.svd(A)
    H = np.eye(3)
    H = Vt[-1].reshape(3,3)
    return H

def displayplot(img,title):
    plt.figure(figsize=(15,15))
    plt.title(title)
    plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
    plt.show()

def RANSAC_alg(f1,f2,matches,nRANSAC,RANSACthresh):
    minMatches = 4
    nBest = 0
    best_inliners = []
    H_estimate = np.eye(3,3)
    global inliner_matchset
    inliner_matchset = []
    for iteration in range(nRANSAC):
        matchSimple = random.sample(matches,minMatches)
        im1_pts = np.empty((minMatches,2))
        im2_pts = np.empty((minMatches,2))
        for i in range(0,minMatches):
            m  = matchSimple[i]
            im1_pts[i] = f1[m.queryIdx].pt
            im2_pts[i] = f2[m.trainIdx].pt

        H_estimate = compute_Hmography(im1_pts,im2_pts)
        inliners = get_inliners(f1,f2,matches,H_estimate,RANSACthresh)
        if len(inliners) > nBest:
            nBest = len(inliners)
            best_inliners= inliners

    print("Number of best inliners", len(best_inliners))
    for i in range(len(best_inliners)):
        inliner_matchset.append(matches[best_inliners[i]])
    im1_pts = np.empty((len(best_inliners),2))
    im2_pts = np.empty((len(best_inliners),2))
    for i in range(0,len(best_inliners)):
        m = inliner_matchset[i]
        im1_pts[i] = f1[m.queryIdx].pt
        im2_pts[i] = f2[m.trainIdx].pt
    M = compute_Hmography(im1_pts,im2_pts)
```

```
    return M, len(best_inliners)
```

In [1]:

```
!pip install opencv-python==3.4.2.17
!pip install opencv-contrib-python==3.4.2.17
```

```
Collecting opencv-python==3.4.2.17
  Downloading opencv_python-3.4.2.17-cp37-cp37m-manylinux1_x86_64.whl (25.0 MB)
     |[redacted]| 25.0 MB 16.0 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-packages (f
rom opencv-python==3.4.2.17) (1.19.5)
Installing collected packages: opencv-python
  Attempting uninstall: opencv-python
    Found existing installation: opencv-python 4.5.1.48
    Uninstalling opencv-python-4.5.1.48:
      Successfully uninstalled opencv-python-4.5.1.48
Successfully installed opencv-python-3.4.2.17
Collecting opencv-contrib-python==3.4.2.17
  Downloading opencv_contrib_python-3.4.2.17-cp37-cp37m-manylinux1_x86_64.whl (30.6 MB)
     |[redacted]| 30.6 MB 15.4 MB/s eta 0:00:01     |[redacted]| 22.5 MB 15.4 MB/s eta 0:00:01     |[redacted]|
24.2 MB 15.4 MB/s eta 0:00:01     |[redacted]| 25.0 MB 15.4 MB/s et
a 0:00:01     |[redacted]| 26.8 MB 15.4 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-packages (f
rom opencv-contrib-python==3.4.2.17) (1.19.5)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-3.4.2.17
```

In [2]:

```
import cv2
cv= cv2.xfeatures2d.SIFT_create()
```

In [6]:

```
files_all = os.listdir('../input/uni-campus-dataset/RGB-img/img/')
files_all.sort()

folder_path = '../input/uni-campus-dataset/RGB-img/img/'
left_files_path_rev = []
right_files_path = []
for file in files_all[:61]:
    left_files_path_rev.append(folder_path + file)


left_files_path = left_files_path_rev[::-1]


for file in files_all[60:100]:
    right_files_path.append(folder_path + file)
```

In [21]:

```
gridsize = 6
clahe = cv2.createCLAHE(clipLimit=2.0,tileGridSize=(gridsize,gridsize))
images_left_bgr = []
images_right_bgr = []
images_left = []
images_right = []


for file in tqdm(left_files_path):
    left_image_sat= cv2.imread(file)
    lab = cv2.cvtColor(left_image_sat, cv2.COLOR_BGR2LAB)
    lab[...,0] = clahe.apply(lab[...,0])
    left_image_sat = cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)
    left_img = cv2.resize(left_image_sat,None,fx=0.35, fy=0.35, interpolation = cv2.INTE
R_AREA)
    images_left.append(cv2.cvtColor(left_img, cv2.COLOR_BGR2GRAY).astype('float32')/255.)
    images_left_bgr.append(left_img)
```

```
for file in tqdm(right_files_path):
    right_image_sat= cv2.imread(file)
    lab = cv2.cvtColor(right_image_sat, cv2.COLOR_BGR2LAB)
    lab[...,0] = clahe.apply(lab[...,0])
    right_image_sat = cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)
    right_img = cv2.resize(right_image_sat,None,fx=0.35,fy=0.35, interpolation = cv2.INT
ER_AREA)
    images_right.append(cv2.cvtColor(right_img, cv2.COLOR_BGR2GRAY).astype('float32')/255
.)
    images_right_bgr.append(right_img)
```

```
100%|██████████| 61/61 [01:06<00:00,  1.10s/it]
100%|██████████| 40/40 [00:43<00:00,  1.09s/it]
```

In [7]:

```
images_left_bgr_no_enhance = []
images_right_bgr_no_enhance = []

for file in tqdm(left_files_path):
    left_image_sat= cv2.imread(file)
    left_img = cv2.resize(left_image_sat,None,fx=0.30, fy=0.30, interpolation = cv2.INTE
R_AREA)
    images_left_bgr_no_enhance.append(left_img)

for file in tqdm(right_files_path):
    right_image_sat= cv2.imread(file)
    right_img = cv2.resize(right_image_sat,None,fx=0.30,fy=0.30, interpolation = cv2.INT
ER_AREA)
    images_right_bgr_no_enhance.append(right_img)
```

```
100%|██████████| 61/61 [00:28<00:00,  2.11it/s]
100%|██████████| 40/40 [00:18<00:00,  2.13it/s]
```

In [9]:

```
Threshl=60;
Octaves=6;
#PatternScales=1.0f;
brisk = cv2.BRISK_create(Threshl,Octaves)

keypoints_all_left_brisk = []
descriptors_all_left_brisk = []
points_all_left_brisk=[]


keypoints_all_right_brisk = []
descriptors_all_right_brisk = []
points_all_right_brisk=[]


for imgs in tqdm(images_left_bgr):
    kpt = brisk.detect(imgs,None)
    kpt,descrip = brisk.compute(imgs, kpt)
    keypoints_all_left_brisk.append(kpt)
    descriptors_all_left_brisk.append(descrip)
    points_all_left_brisk.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr):
    kpt = brisk.detect(imgs,None)
    kpt,descrip = brisk.compute(imgs, kpt)
    keypoints_all_right_brisk.append(kpt)
    descriptors_all_right_brisk.append(descrip)
    points_all_right_brisk.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

```
100%|██████████| 61/61 [00:41<00:00,  1.47it/s]
100%|██████████| 40/40 [00:25<00:00,  1.54it/s]
```

In [9]:

```
orb = cv2.ORB_create(5000)
keypoints_all_left_orb = []
descriptors_all_left_orb = []
points_all_left_orb=[]

keypoints_all_right_orb = []
descriptors_all_right_orb = []
points_all_right_orb=[]

for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = orb.detect(imgs,None)
    kpt,descrip = orb.compute(imgs, kpt)
    keypoints_all_left_orb.append(kpt)
    descriptors_all_left_orb.append(descrip)
    points_all_left_orb.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = orb.detect(imgs,None)
    kpt,descrip = orb.compute(imgs, kpt)
    keypoints_all_right_orb.append(kpt)
    descriptors_all_right_orb.append(descrip)
    points_all_right_orb.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

```
100%|██████████| 61/61 [00:06<00:00,  9.17it/s]
100%|██████████| 40/40 [00:04<00:00,  9.79it/s]
```

In [ ]:

```
kaze = cv2.KAZE_create()
keypoints_all_left_kaze = []
descriptors_all_left_kaze = []
points_all_left_kaze=[]

keypoints_all_right_kaze = []
descriptors_all_right_kaze = []
points_all_right_kaze=[]

for imgs in tqdm(images_left_bgr):
    kpt = kaze.detect(imgs,None)
    kpt,descrip = kaze.compute(imgs, kpt)
    keypoints_all_left_kaze.append(kpt)
    descriptors_all_left_kaze.append(descrip)
    points_all_left_kaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr):
    kpt = kaze.detect(imgs,None)
    kpt,descrip = kaze.compute(imgs, kpt)
    keypoints_all_right_kaze.append(kpt)
    descriptors_all_right_kaze.append(descrip)
    points_all_right_kaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [8]:

```
tqdm = partial(tqdm, position=0, leave=True)
```

In [ ]:

```
akaze = cv2.AKAZE_create()
keypoints_all_left_akaze = []
descriptors_all_left_akaze = []
points_all_left_akaze=[]

keypoints_all_right_akaze = []
descriptors_all_right_akaze = []
points_all_right_akaze=[]

for imgs in tqdm(images_left_bgr):
    kpt = akaze.detect(imgs,None)
    kpt,descrip = akaze.compute(imgs, kpt)
    keypoints_all_left_akaze.append(kpt)
    descriptors_all_left_akaze.append(descrip)
```

```
    points_all_left_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
for imgs in tqdm(images_right_bgr):
    kpt = akaze.detect(imgs,None)
    kpt,descrip = akaze.compute(imgs, kpt)
    keypoints_all_right_akaze.append(kpt)
    descriptors_all_right_akaze.append(descrip)
    points_all_right_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

```
star = cv2.xfeatures2d.StarDetector_create()
brief = cv2.xfeatures2d.BriefDescriptorExtractor_create()
keypoints_all_left_star = []
descriptors_all_left_brief = []
points_all_left_star=[]


keypoints_all_right_star = []
descriptors_all_right_brief = []
points_all_right_star=[]

for imgs in tqdm(images_left_bgr):
    kpt = star.detect(imgs,None)
    kpt,descrip = brief.compute(imgs, kpt)
    keypoints_all_left_star.append(kpt)
    descriptors_all_left_brief.append(descrip)
    points_all_left_star.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr):
    kpt = star.detect(imgs,None)
    kpt,descrip = brief.compute(imgs, kpt)
    keypoints_all_right_star.append(kpt)
    descriptors_all_right_brief.append(descrip)
    points_all_right_star.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

```
Threshl=60;
Octaves=8;
#PatternScales=1.0f;
brisk = cv2.BRISK_create(Threshl,Octaves)
freak = cv2.xfeatures2d.FREAK_create()
keypoints_all_left_freak = []
descriptors_all_left_freak = []
points_all_left_freak=[]


keypoints_all_right_freak = []
descriptors_all_right_freak = []
points_all_right_freak=[]

for imgs in tqdm(images_left_bgr):
    kpt = brisk.detect(imgs)
    kpt,descrip = freak.compute(imgs, kpt)
    keypoints_all_left_freak.append(kpt)
    descriptors_all_left_freak.append(descrip)
    points_all_left_freak.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr):
    kpt = brisk.detect(imgs,None)
    kpt,descrip = freak.compute(imgs, kpt)
    keypoints_all_right_freak.append(kpt)
    descriptors_all_right_freak.append(descrip)
    points_all_right_freak.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

```
mser = cv2.MSER_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_mser = []
descriptors_all_left_mser = []
```

```
points_all_left_mser=[]

keypoints_all_right_mser = []
descriptors_all_right_mser = []
points_all_right_mser=[]
for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = mser.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_mser.append(kpt)
    descriptors_all_left_mser.append(descrip)
    points_all_left_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))


for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = mser.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_mser.append(kpt)
    descriptors_all_right_mser.append(descrip)
    points_all_right_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

```
agast = cv2.AgastFeatureDetector_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_agast = []
descriptors_all_left_agast = []
points_all_left_agast=[]

keypoints_all_right_agast = []
descriptors_all_right_agast = []
points_all_right_agast=[]


for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = agast.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_agast.append(kpt)
    descriptors_all_left_agast.append(descrip)
    points_all_left_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))


for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = agast.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_agast.append(kpt)
    descriptors_all_right_agast.append(descrip)
    points_all_right_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [10]:

```
fast = cv2.FastFeatureDetector_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_fast = []
descriptors_all_left_fast = []
points_all_left_fast=[]

keypoints_all_right_fast = []
descriptors_all_right_fast = []
points_all_right_fast=[]
for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = fast.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_fast.append(kpt)
    descriptors_all_left_fast.append(descrip)
    points_all_left_fast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = fast.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_fast.append(kpt)
    descriptors_all_right_fast.append(descrip)
```

```
    points_all_right_fast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

```
100%|██████████| 61/61 [04:32<00:00,  4.46s/it]
100%|██████████| 40/40 [03:13<00:00,  4.84s/it]
```

In [ ]:

```
gftt = cv2.GFTTDetector_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_gftt = []
descriptors_all_left_gftt = []
points_all_left_gftt=[]

keypoints_all_right_gftt = []
descriptors_all_right_gftt = []
points_all_right_gftt=[]
for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = gftt.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_gftt.append(kpt)
    descriptors_all_left_gftt.append(descrip)
    points_all_left_gftt.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = gftt.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_gftt.append(kpt)
    descriptors_all_right_gftt.append(descrip)
    points_all_right_gftt.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

```
daisy = cv2.xfeatures2d.DAISY_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_daisy = []
descriptors_all_left_daisy = []
points_all_left_daisy=[]

keypoints_all_right_daisy = []
descriptors_all_right_daisy = []
points_all_right_daisy=[]

for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = sift.detect(imgs,None)
    kpt,descrip = daisy.compute(imgs, kpt)
    keypoints_all_left_daisy.append(kpt)
    descriptors_all_left_daisy.append(descrip)
    points_all_left_daisy.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = sift.detect(imgs,None)
    kpt,descrip = daisy.compute(imgs, kpt)
    keypoints_all_right_daisy.append(kpt)
    descriptors_all_right_daisy.append(descrip)
    points_all_right_daisy.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [13]:

```
surf = cv2.xfeatures2d.SURF_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_surfsift = []
descriptors_all_left_surfsift = []
points_all_left_surfsift=[]

keypoints_all_right_surfsift = []
descriptors_all_right_surfsift = []
points_all_right_surfsift=[]

for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = surf.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
```

```
    keypoints_all_left_surfsift.append(kpt)
    descriptors_all_left_surfsift.append(descrip)
    points_all_left_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = surf.detect(imgs,None)

    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_surfsift.append(kpt)
    descriptors_all_right_surfsift.append(descrip)
    points_all_right_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

```
100%|██████████| 61/61 [14:21<00:00, 14.11s/it]
100%|██████████| 40/40 [09:37<00:00, 14.44s/it]
```

In [ ]:

```
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_sift = []
descriptors_all_left_sift = []
points_all_left_sift=[]

keypoints_all_right_sift = []
descriptors_all_right_sift = []
points_all_right_sift=[]


for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = sift.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_sift.append(kpt)
    descriptors_all_left_sift.append(descrip)
    points_all_left_sift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = sift.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_sift.append(kpt)
    descriptors_all_right_sift.append(descrip)
    points_all_right_sift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

```
surf = cv2.xfeatures2d.SURF_create()
keypoints_all_left_surf = []
descriptors_all_left_surf = []
points_all_left_surf=[]

keypoints_all_right_surf = []
descriptors_all_right_surf = []
points_all_right_surf=[]
for imgs in tqdm(images_left_bgr):
    kpt = surf.detect(imgs,None)
    kpt,descrip = surf.compute(imgs, kpt)
    keypoints_all_left_surf.append(kpt)
    descriptors_all_left_surf.append(descrip)
    points_all_left_surf.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for imgs in tqdm(images_right_bgr):
    kpt = surf.detect(imgs,None)
    kpt,descrip = surf.compute(imgs, kpt)
    keypoints_all_right_surf.append(kpt)
    descriptors_all_right_surf.append(descrip)
    points_all_right_surf.append(np.asarray([[p.pt[0],p.pt[1]] for p in kpt]))
```

In [ ]:

```
# sift  = cv2.xfeatures2d.SURF_Create()
# keypoints_all_left_surf = []
# descriptor_all_left_surf = []
# points_all_left_surf = []
```

```
# keypoints_all_right_surf = []
# descriptor_all_right_surf = []
# points_all_right_surf = []

#  for images in tqdm(left_images_bgr):
#  kpt = surf.detect(imgs,None)
# kpt, descrip = surf.compute(imgs,kpt)
# keypoints_all_left_surf.append(kpt)
#descriptor_all_left_surf.append(descrip)
#points_all_left_surf.append(np.asarray([[p.pt[0],p.pt[1]] for p in kpt])
#  points_all_left_surf.append(np.asarray([[p.pt[0], p.pt[1] for p in  kpt]]))
```

In [ ]:

```
class RootSIFT:
    def __init__(self):
        # initialize the SIFT feature extractor
        #self.extractor = cv2.DescriptorExtractor_create("SIFT")
        self.sift = cv2.xfeatures2d.SIFT_create()
    def compute(self, image, kps, eps=1e-7):
        # compute SIFT descriptors
        (kps, descs) = self.sift.compute(image, kps)
        # if there are no keypoints or descriptors, return an empty tuple
        if len(kps) == 0:
            return ([], None)
        # apply the Hellinger kernel by first L1-normalizing, taking the
        # square-root, and then L2-normalizing
        descs /= (np.linalg.norm(descs, axis=0, ord=2) + eps)
        descs /= (descs.sum(axis=0) + eps)
        descs = np.sqrt(descs)
        #descs /= (np.linalg.norm(descs, axis=0, ord=2) + eps)
        # return a tuple of the keypoints and descriptors
        return (kps, descs)
```

In [ ]:

```
sift = cv2.xfeatures2d.SIFT_create()
rootsift = RootSIFT()
keypoints_all_left_rootsift = []
descriptors_all_left_rootsift = []
points_all_left_rootsift=[]

keypoints_all_right_rootsift = []
descriptors_all_right_rootsift = []
points_all_right_rootsift=[]

for imgs in tqdm(images_left_bgr):
    kpt = sift.detect(imgs,None)
    kpt,descrip = rootsift.compute(imgs, kpt)
    keypoints_all_left_rootsift.append(kpt)
    descriptors_all_left_rootsift.append(descrip)
    points_all_left_rootsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
for imgs in tqdm(images_right_bgr):
    kpt = sift.detect(imgs,None)
    kpt,descrip = rootsift.compute(imgs, kpt)
    keypoints_all_right_rootsift.append(kpt)
    descriptors_all_right_rootsift.append(descrip)
    points_all_right_rootsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [11]:

```
!git clone https://github.com/magicleap/SuperPointPretrainedNetwork.git
```

```
Cloning into 'SuperPointPretrainedNetwork'...
remote: Enumerating objects: 81, done.
remote: Total 81 (delta 0), reused 0 (delta 0), pack-reused 81
Unpacking objects: 100% (81/81), done.
```

In [12]:

```
weights_path = 'SuperPointPretrainedNetwork/superpoint_v1.pth'
```

```
weights_path = 'SuperPointPretrainedNetwork/superpoint_v1.pth'
cuda = 'True'
```

In [13]:

```
def to_kpts(pts,size=1):
    return [cv2.KeyPoint(pt[0],pt[1],size) for pt in pts]
```

In [14]:

```
torch.cuda.empty_cache()
class SuperPointNet(nn.Module):
    def __init__(self):
        super(SuperPointNet,self).__init__()
        self.relu = nn.ReLU(inplace=True)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)
        c1,c2,c3,c4,c5,d1 = 64,64,128,128,256,256
        self.conv1a = nn.Conv2d(1,c1,kernel_size=3,stride=1,padding=1)
        self.conv1b = nn.Conv2d(c1,c1,kernel_size=3,stride=1,padding=1)
        self.conv2a = nn.Conv2d(c1,c2,kernel_size=3,stride=1,padding=1)
        self.conv2b = nn.Conv2d(c2,c2,kernel_size=3,stride=1,padding=1)
        self.conv3a = nn.Conv2d(c2,c3,kernel_size=3,stride=1,padding=1)
        self.conv3b = nn.Conv2d(c3,c3,kernel_size=3,stride=1,padding=1)
        self.conv4a = nn.Conv2d(c3,c4,kernel_size=3,stride=1,padding=1)
        self.conv4b = nn.Conv2d(c4,c4,kernel_size=3,stride=1,padding=1)
        self.convPa = nn.Conv2d(c4,c5,kernel_size=3,stride=1,padding=1)
        self.convPb = nn.Conv2d(c5,65,kernel_size=1,stride=1,padding=0)
        self.convDa = nn.Conv2d(c4,c5,kernel_size=3,stride=1,padding=1)


        self.convDb = nn.Conv2d(c5,d1,kernel_size=1,stride=1,padding=0)

    def forward(self,x):
        x = self.relu(self.conv1a(x))
        x = self.relu(self.conv1b(x))
        x = self.pool(x)
        x = self.relu(self.conv2a(x))
        x = self.relu(self.conv2b(x))
        x = self.pool(x)
        x  = self.relu(self.conv3a(x))
        x = self.relu(self.conv3b(x))
        x = self.pool(x)
        x = self.relu(self.conv4a(x))
        x = self.relu(self.conv4b(x))
        cPa = self.relu(self.convPa(x))
        semi = self.convPb(cPa)
        cDa = self.relu(self.convDa(x))
        desc = self.convDb(cDa)
        dn = torch.norm(desc,p=2,dim=1)
        desc = desc.div(torch.unsqueeze(dn,1))
        return semi,desc


class SuperPointFrontend(object):
    def __init__(self,weights_path,nms_dist,conf_thresh, nn_thresh,cuda=True):
        self.name = 'SuperPoint'
        self.cuda = cuda
        self.nms_dist = nms_dist
        self.conf_thresh = conf_thresh
        self.nn_thresh = nn_thresh
        self.cell = 8
        self.border_remove = 4

        self.net = SuperPointNet()
        if cuda:
            self.net.load_state_dict(torch.load(weights_path))
            self.net = self.net.cuda()
        else:
            self.net.load_state_dict(torch.load(weights_path,map_location=lambda storage
, loc: storage))
        self.net.eval()
```

```
    def nms_fast(self,in_corners,H,W,dist_thresh):
        grid = np.zeros((H,W)).astype(int)
        inds = np.zeros((H,W)).astype(int)
        inds1 = np.argsort(-in_corners[2,:])
        corners  = in_corners[:,inds1]
        rcorners = corners[:2,:].round().astype(int)
        if rcorners.shape[1] == 0:
            return np.zeros((3,0)).astype(int), np.zeros(0).astype(int)
        if rcorners.shape[1] == 1:
            out =  np.vstack((rcorners,in_corners[2])).reshape(3,1)
            return out,np.zeros((1)).astype(int)
        for i, rc in enumerate(rcorners.T):
            grid[rcorners[1,i],rcorners[0,i]] =1
            inds[rcorners[1,i],rcorners[0,i]] =i
        pad = dist_thresh
        grid = np.pad(grid,((pad,pad),(pad,pad)),mode='constant')
        count = 0
        for i,rc  in enumerate(rcorners.T):
            pt = (rc[0]+pad, rc[1]+pad)
            if grid[pt[1], pt[0]] == 1:
                grid[pt[1]-pad:pt[1]+pad+1, pt[0]-pad:pt[0]+pad+1]=0


                grid[pt[1], pt[0]] = -1
                count += 1

        keepy, keepx = np.where(grid==-1)
        keepy,keepx = keepy-pad , keepx-pad
        inds_keep = inds[keepy, keepx]
        out  = corners[:,inds_keep]
        values = out[-1,:]
        inds2 = np.argsort(-values)
        out = out[:,inds2]
        out_inds = inds1[inds_keep[inds2]]
        return out, out_inds

    def run(self,img):
        assert img.ndim == 2
        assert img.dtype == np.float32
        H,W = img.shape[0], img.shape[1]
        inp = img.copy()
        inp = (inp.reshape(1,H,W))
        inp = torch.from_numpy(inp)
        inp = torch.autograd.Variable(inp).view(1,1,H,W)
        if self.cuda:
            inp = inp.cuda()
        outs = self.net.forward(inp)
        semi,coarse_desc = outs[0],outs[1]
        semi = semi.data.cpu().numpy().squeeze()


        dense = np.exp(semi)
        dense = dense / (np.sum(dense,axis=0)+.00001)
        nodust = dense[:-1,:,:]
        Hc = int(H / self.cell)
        Wc = int(W  / self.cell)
        nodust = np.transpose(nodust,[1,2,0])
        heatmap = np.reshape(nodust,[Hc,Wc,self.cell,self.cell])
        heatmap = np.transpose(heatmap,[0,2,1,3])
        heatmap = np.reshape(heatmap,[Hc*self.cell, Wc*self.cell])
        prob_map  = heatmap/np.sum(np.sum(heatmap))

        return heatmap,coarse_desc

    def key_pt_sampling(self,img,heat_map,coarse_desc,sampled):
        H,W = img.shape[0], img.shape[1]
        xs,ys = np.where(heat_map >= self.conf_thresh)
        if len(xs) == 0:
            return np.zeros((3,0)),None,None
        print("Number of pts selected:",len(xs))

        pts = np.zeros((3,len(xs)))
```

```
        pts[0,:] = ys
        pts[1,:] = xs
        pts[2,:] = heat_map[xs,ys]
        pts,_ = self.nms_fast(pts,H,W,dist_thresh=self.nms_dist)
        inds = np.argsort(pts[2,:])
        pts = pts[:,inds[::-1]]
        bord = self.border_remove
        toremoveW = np.logical_or(pts[0,:] < bord, pts[0,:] >= (W-bord))
        toremoveH = np.logical_or(pts[1,:] < bord, pts[0,:] >= (H-bord))
        toremove = np.logical_or(toremoveW, toremoveH)
        pts = pts[:,~toremove]
        pts = pts[:,0:sampled]
        D = coarse_desc.shape[1]
        if pts.shape[1] == 0:
            desc = np.zeros((D,0))
        else:
            samp_pts = torch.from_numpy(pts[:2,:].copy())
            samp_pts[0,:] = (samp_pts[0,:] / (float(W)/2.))-1.
            samp_pts[1,:] = (samp_pts[1,:] / (float(W)/2.))-1.
            samp_pts = samp_pts.transpose(0,1).contiguous()
            samp_pts = samp_pts.view(1,1,-1,2)
            samp_pts = samp_pts.float()
            if self.cuda:
                samp_pts = samp_pts.cuda()
            desc = nn.functional.grid_sample(coarse_desc, samp_pts)
            desc = desc.data.cpu().numpy().reshape(D,-1)
            desc /= np.linalg.norm(desc,axis=0)[np.newaxis,:]
        return pts,desc
```

In [15]:

```
print('Load pre trained network')
fe = SuperPointFrontend(weights_path = weights_path, nms_dist = 4, conf_thresh  = 0.015,
nn_thresh=0.7,
                        cuda = cuda)
print('Successfully  loaded pretrained network')
```

```
Load pre trained network
Successfully  loaded pretrained network
```

In [ ]:

```
keypoint_all_left_superpoint = []
descriptor_all_left_superpoint = []
point_all_left_superpoint = []

keypoints_all_right_superpoint = []
descriptors_all_right_superpoint = []
points_all_right_superpoint = []

for ifpth in tqdm(images_left):
    heatmap1, coarse_desc1 = fe.run(ifpth)
    pts_1, desc_1 = fe.key_pt_sampling(ifpth,heatmap1,coarse_desc1,2000)

    keypoint_all_left_superpoint.append(to_kpts(pts_1.T))
    descriptor_all_left_superpoint.append(desc_1.T)
    point_all_left_superpoint.append(pts_1.T)





for rfpth in tqdm(images_right):
    heatmap1, coarse_desc1 = fe.run(rfpth)
    pts_1, desc_1 = fe.key_pt_sampling(rfpth,heatmap1,coarse_desc1,2000)

    keypoints_all_right_superpoint.append(to_kpts(pts_1.T))
    descriptors_all_right_superpoint.append(desc_1.T)
    points_all_right_superpoint.append(pts_1.T)
```

In [ ]:

```
num_kps_superpoint = []
for j in tqdm(keypoint_all_left_superpoint + keypoints_all_right_superpoint):
    num_kps_superpoint.append(len(j))
```

In [16]:

```
num_kps_brisk = []
for j in tqdm(keypoints_all_left_brisk + keypoints_all_right_brisk):
    num_kps_brisk.append(len(j))
```

```
100%|██████████| 101/101 [00:00<00:00, 206243.77it/s]
```

In [16]:

```
num_kps_orb = []
for j  in tqdm(keypoints_all_left_orb + keypoints_all_right_orb):
    num_kps_orb.append(len(j))
```

```
100%|██████████| 101/101 [00:00<00:00, 378236.34it/s]
```

In [16]:

```
num_kps_fast = []
for j in tqdm(keypoints_all_left_fast + keypoints_all_right_fast):
    num_kps_fast.append(len(j))
```

```
100%|██████████| 101/101 [00:00<00:00, 303673.62it/s]
```

In [ ]:

```
num_kps_kaze = []
for j  in tqdm(keypoints_all_left_kaze + keypoints_all_right_kaze):
    num_kps_kaze.append(len(j))
```

In [ ]:

```
num_kps_akaze  = []




for j in tqdm(keypoints_all_left_akaze + keypoints_all_right_akaze):
    num_kps_akaze.append(len(j))
```

In [ ]:

```
num_kps_freak = []
for j in tqdm(keypoints_all_left_freak + keypoints_all_right_freak):
    num_kps_freak.append(len(j))
```

In [ ]:

```
num_kps_mser =[]
for j in tqdm(keypoints_all_left_mser + keypoints_all_right_mser):
    num_kps_mser.append(len(j))
```

In [ ]:

```
num_kps_gftt =[]
for j in tqdm(keypoints_all_left_gftt + keypoints_all_right_gftt):
    num_kps_gftt.append(len(j))
```

In [ ]:

```
num_kps_daisy = []
for  j in tqdm(keypoints_all_left_daisy + keypoints_all_right_daisy):
    num_kps_daisy.append(j)
```

In [ ]:

```
num_kps_star = []
for  j in tqdm(keypoints_all_left_star + keypoints_all_right_star):
    num_kps_star.append(len(j))
```

In [ ]:

```
num_kps_sift = []
for j in tqdm(keypoints_all_left_sift + keypoints_all_right_sift):
    num_kps_sift.append(len(j))
```

In [ ]:

```
num_kps_surf = []
for j in tqdm(keypoints_all_left_surf + keypoints_all_right_surf):
    num_kps_surf.append(len(j))
```

In [19]:

```
num_kps_surfsift = []
for j in tqdm(keypoints_all_left_surfsift + keypoints_all_right_surfsift):
    num_kps_surfsift.append(len(j))
```

```
100%|██████████| 101/101 [00:00<00:00, 211600.75it/s]
```

In [ ]:

```
num_kps_agast  = []
for j in tqdm(keypoints_all_left_agast + keypoints_all_right_agast):
    num_kps_agast.append(len(j))
```

In [17]:

```
def compute_homography_fast(matched_pts1, matched_pts2,thresh=4):
    #matched_pts1 = cv2.KeyPoint_convert(matched_kp1)
    #matched_pts2 = cv2.KeyPoint_convert(matched_kp2)
    # Estimate the homography between the matches using RANSAC
    H, inliers = cv2.findHomography(matched_pts1,matched_pts2,cv2.RANSAC, ransacReprojTh
reshold =thresh)
    inliers = inliers.flatten()
    return H, inliers
```

In [18]:

```
def get_Hmatrix(imgs,keypts,pts,descripts,ratio=0.8,thresh=4,disp=False):
    FLANN_INDEX_KDTREE = 2
    index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
    search_params = dict(checks=50)
    flann = cv2.FlannBasedMatcher(index_params, search_params)
    #flann = cv2.BFMatcher()
    lff1 = np.float32(descripts[0])
    lff = np.float32(descripts[1])
    matches_lf1_lf = flann.knnMatch(lff1, lff, k=2)
    print("\nNumber of matches",len(matches_lf1_lf))
    matches_4 = []
    ratio = ratio
    # loop over the raw matches
    for m in matches_lf1_lf:
        # ensure the distance is within a certain ratio of each
        # other (i.e. Lowe’s ratio test)
        if len(m) == 2 and m[0].distance < m[1].distance * ratio:

            matches_4.append(m[0])
    print("Number of matches After Lowe's Ratio",len(matches_4))
    matches_idx = np.array([m.queryIdx for m in matches_4])
    imm1_pts = np.array([keypts[0][idx].pt for idx in matches_idx])
    matche_idx = np.array([m.trainIdx for m in matches_4])
    imm2_pts = np.array([keypts[1][idx].pt for idx in matche_idx])

    '''
    # Estimate homography 1
    #Compute H1
```

```
    # Estimate homography 1
    #Compute H1
    imm1_pts=np.empty((len(matches_4),2))
    imm2_pts=np.empty((len(matches_4),2))
    for i in range(0,len(matches_4)):
    m = matches_4[i]
    (a_x, a_y) = keypts[0][m.queryIdx].pt
    (b_x, b_y) = keypts[1][m.trainIdx].pt
    imm1_pts[i]=(a_x, a_y)
    imm2_pts[i]=(b_x, b_y)
    H=compute_Homography(imm1_pts,imm2_pts)
    #Robustly estimate Homography 1 using RANSAC
    Hn, best_inliers=RANSAC_alg(keypts[0] ,keypts[1], matches_4, nRANSAC=1000, RANSACthre
sh=6)
    '''
    Hn,inliers = compute_homography_fast(imm1_pts,imm2_pts)

    inlier_matchset = np.array(matches_4)[inliers.astype(bool)].tolist()
    print("Number of Robust matches",len(inlier_matchset))
    print("\n")
    '''
    if len(inlier_matchset)<50:
        matches_4 = []
        ratio = 0.67
        # loop over the raw matches
        for m in matches_lf1_lf:
           # ensure the distance is within a certain ratio of each
           # other (i.e. Lowe's ratio test)
           if len(m) == 2 and m[0].distance < m[1].distance * ratio:
           #matches_1.append((m[0].trainIdx, m[0].queryIdx))
           matches_4.append(m[0])
        print("Number of matches After Lowe's Ratio New",len(matches_4))
        matches_idx = np.array([m.queryIdx for m in matches_4])
        imm1_pts = np.array([keypts[0][idx].pt for idx in matches_idx])
        matches_idx = np.array([m.trainIdx for m in matches_4])
        imm2_pts = np.array([keypts[1][idx].pt for idx in matches_idx])
        Hn,inliers = compute_homography_fast_other(imm1_pts,imm2_pts)
        inlier_matchset = np.array(matches_4)[inliers.astype(bool)].tolist()
        print("Number of Robust matches New",len(inlier_matchset))
        print("\n")
       '''
    #H=compute_Homography(imm1_pts,imm2_pts)
    #Robustly estimate Homography 1 using RANSAC
    #Hn=RANSAC_alg(keypts[0] ,keypts[1], matches_4, nRANSAC=1500, RANSACthresh=6)
    #global inlier_matchset
    if disp==True:
        dispimg1=cv2.drawMatches(imgs[0], keypts[0], imgs[1], keypts[1], inlier_matchset
, None,flags=2)
        displayplot(dispimg1,'Robust Matching between Reference Image and Right Image ')
    return Hn/Hn[2,2], len(matches_lf1_lf), len(inlier_matchset)
```

In [19]:

```
from functools import partial
from tqdm import tqdm
tqdm = partial(tqdm, position=0, leave=True)
```

In [20]:

```
H_left_brisk = []
H_right_brisk = []

num_matches_brisk = []
num_good_matches_brisk = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_brisk[j:j+2][::-1],points_all_left_brisk[j:j+2][::-1],descriptors_all_left_brisk[j:j+2]
```

```
[::-1])
    H_left_brisk.append(H_a)
    num_matches_brisk.append(matches)
    num_good_matches_brisk.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_brisk[j:j+2][::-1],points_all_right_brisk[j:j+2][::-1],descriptors_all_right_brisk[j:
j+2][::-1])
    H_right_brisk.append(H_a)
    num_matches_brisk.append(matches)
    num_good_matches_brisk.append(gd_matches)
```

```
  2%|          | 1/61 [00:01<01:02,  1.04s/it]

Number of matches 17087
Number of matches After Lowe's Ratio 837
Number of Robust matches 347


  3%|          | 2/61 [00:02<01:13,  1.25s/it]

Number of matches 22471
Number of matches After Lowe's Ratio 630
Number of Robust matches 222


  5%|          | 3/61 [00:04<01:21,  1.41s/it]

Number of matches 18668
Number of matches After Lowe's Ratio 360
Number of Robust matches 16


  7%|          | 4/61 [00:05<01:16,  1.34s/it]

Number of matches 16606
Number of matches After Lowe's Ratio 1623
Number of Robust matches 1006


  8%|          | 5/61 [00:06<01:12,  1.29s/it]

Number of matches 19717
Number of matches After Lowe's Ratio 1870
Number of Robust matches 1065


 10%|          | 6/61 [00:07<01:12,  1.32s/it]

Number of matches 18849
Number of matches After Lowe's Ratio 1609
Number of Robust matches 785


 11%|          | 7/61 [00:09<01:16,  1.42s/it]

Number of matches 22695
Number of matches After Lowe's Ratio 1939
Number of Robust matches 1121


 13%|          | 8/61 [00:11<01:19,  1.49s/it]

Number of matches 16338
```

```
Number of matches After Lowe's Ratio 976
Number of Robust matches 549
```

```
 15%|          | 9/61 [00:12<01:20,  1.54s/it]
```

```
Number of matches 23595
Number of matches After Lowe's Ratio 1699
Number of Robust matches 1059
```

```
 16%|          | 10/61 [00:14<01:20,  1.59s/it]
```

```
Number of matches 20279
Number of matches After Lowe's Ratio 1147
Number of Robust matches 679
```

```
 18%|          | 11/61 [00:16<01:20,  1.60s/it]
```

```
Number of matches 25716
Number of matches After Lowe's Ratio 2148
Number of Robust matches 1612
```

```
 20%|          | 12/61 [00:18<01:24,  1.72s/it]
```

```
Number of matches 25252
Number of matches After Lowe's Ratio 2404
Number of Robust matches 1716
```

```
 21%|          | 13/61 [00:20<01:25,  1.79s/it]
```

```
Number of matches 28588
Number of matches After Lowe's Ratio 2307
Number of Robust matches 1621
```

```
 23%|          | 14/61 [00:22<01:33,  2.00s/it]
```

```
Number of matches 28948
Number of matches After Lowe's Ratio 3499
Number of Robust matches 2828
```

```
 25%|          | 15/61 [00:24<01:37,  2.11s/it]
```

```
Number of matches 26122
Number of matches After Lowe's Ratio 2685
Number of Robust matches 1824
```

```
 26%|          | 16/61 [00:26<01:32,  2.05s/it]
```

```
Number of matches 22890
Number of matches After Lowe's Ratio 2579
Number of Robust matches 1984
```

```
 28%|          | 17/61 [00:28<01:25,  1.94s/it]
```

```
Number of matches 22468
Number of matches After Lowe's Ratio 2356
Number of Robust matches 1855
```

```
 30%|[redacted]        | 18/61 [00:30<01:19,  1.85s/it]

Number of matches 23015
Number of matches After Lowe's Ratio 2622
Number of Robust matches 1818


 31%|[redacted]        | 19/61 [00:32<01:20,  1.92s/it]

Number of matches 21747
Number of matches After Lowe's Ratio 2904
Number of Robust matches 2169


 33%|[redacted]        | 20/61 [00:33<01:14,  1.81s/it]

Number of matches 21215
Number of matches After Lowe's Ratio 2115
Number of Robust matches 1458


 34%|[redacted]        | 21/61 [00:35<01:09,  1.73s/it]

Number of matches 23009
Number of matches After Lowe's Ratio 1836
Number of Robust matches 1030


 36%|[redacted]        | 22/61 [00:36<01:06,  1.72s/it]

Number of matches 22867
Number of matches After Lowe's Ratio 1985
Number of Robust matches 1253


 38%|[redacted]        | 23/61 [00:38<01:06,  1.75s/it]

Number of matches 23813
Number of matches After Lowe's Ratio 2191
Number of Robust matches 1446


 39%|[redacted]        | 24/61 [00:40<01:05,  1.76s/it]

Number of matches 25297
Number of matches After Lowe's Ratio 1855
Number of Robust matches 1250


 41%|[redacted]        | 25/61 [00:42<01:09,  1.94s/it]

Number of matches 32196
Number of matches After Lowe's Ratio 2213
Number of Robust matches 844


 43%|[redacted]        | 26/61 [00:45<01:19,  2.27s/it]

Number of matches 26944
Number of matches After Lowe's Ratio 1794
Number of Robust matches 857


 44%|[redacted]        | 27/61 [00:48<01:16,  2.24s/it]

Number of matches 23440
```

```
Number of matches After Lowe's Ratio 1765
Number of Robust matches 997


 46%|[redacted]      | 28/61 [00:49<01:08,  2.08s/it]

Number of matches 20403
Number of matches After Lowe's Ratio 1416
Number of Robust matches 672


 48%|[redacted]      | 29/61 [00:51<01:01,  1.93s/it]

Number of matches 24190
Number of matches After Lowe's Ratio 943
Number of Robust matches 391


 49%|[redacted]      | 30/61 [00:53<01:00,  1.94s/it]

Number of matches 23707
Number of matches After Lowe's Ratio 1224
Number of Robust matches 563


 51%|[redacted]      | 31/61 [00:55<00:58,  1.94s/it]

Number of matches 24206
Number of matches After Lowe's Ratio 626
Number of Robust matches 237


 52%|[redacted]      | 32/61 [00:57<00:54,  1.87s/it]

Number of matches 17116
Number of matches After Lowe's Ratio 338
Number of Robust matches 58


 54%|[redacted]      | 33/61 [00:58<00:46,  1.66s/it]

Number of matches 16368
Number of matches After Lowe's Ratio 1145
Number of Robust matches 684


 56%|[redacted]      | 34/61 [00:59<00:39,  1.48s/it]

Number of matches 13895
Number of matches After Lowe's Ratio 1210
Number of Robust matches 686


 57%|[redacted]      | 35/61 [01:00<00:34,  1.31s/it]

Number of matches 17372
Number of matches After Lowe's Ratio 1108
Number of Robust matches 633


 59%|[redacted]      | 36/61 [01:01<00:33,  1.33s/it]

Number of matches 21884
Number of matches After Lowe's Ratio 1536
Number of Robust matches 710
```

```
 61%|▌▌▌▌▌▌    | 37/61 [01:03<00:34,  1.44s/it]

Number of matches 31581
Number of matches After Lowe's Ratio 1544
Number of Robust matches 600


 62%|▌▌▌▌▌▌▌   | 38/61 [01:06<00:43,  1.88s/it]

Number of matches 34293
Number of matches After Lowe's Ratio 2153
Number of Robust matches 620


 64%|▌▌▌▌▌▌▌   | 39/61 [01:09<00:48,  2.22s/it]

Number of matches 30683
Number of matches After Lowe's Ratio 2065
Number of Robust matches 773


 66%|▌▌▌▌▌▌▌   | 40/61 [01:11<00:47,  2.27s/it]

Number of matches 24337
Number of matches After Lowe's Ratio 2009
Number of Robust matches 994


 67%|▌▌▌▌▌▌▌   | 41/61 [01:13<00:42,  2.14s/it]

Number of matches 23240
Number of matches After Lowe's Ratio 2362
Number of Robust matches 1465


 69%|▌▌▌▌▌▌▌   | 42/61 [01:15<00:38,  2.03s/it]

Number of matches 22027
Number of matches After Lowe's Ratio 2408
Number of Robust matches 1744


 70%|▌▌▌▌▌▌▌▌  | 43/61 [01:17<00:38,  2.15s/it]

Number of matches 22221
Number of matches After Lowe's Ratio 2436
Number of Robust matches 1622


 72%|▌▌▌▌▌▌▌▌  | 44/61 [01:19<00:34,  2.00s/it]

Number of matches 26810
Number of matches After Lowe's Ratio 2364
Number of Robust matches 1455


 74%|▌▌▌▌▌▌▌▌  | 45/61 [01:21<00:32,  2.04s/it]

Number of matches 28729
Number of matches After Lowe's Ratio 2922
Number of Robust matches 1681


 75%|▌▌▌▌▌▌▌▌  | 46/61 [01:23<00:31,  2.12s/it]

Number of matches 27417
```

```
Number of matches 27417
Number of matches After Lowe's Ratio 2843
Number of Robust matches 1759

 77%|[redacted]  | 47/61 [01:26<00:30,  2.17s/it]

Number of matches 28921
Number of matches After Lowe's Ratio 2731
Number of Robust matches 1361

 79%|[redacted]  | 48/61 [01:28<00:29,  2.26s/it]

Number of matches 24951
Number of matches After Lowe's Ratio 1969
Number of Robust matches 1223

 80%|[redacted]  | 49/61 [01:30<00:25,  2.14s/it]

Number of matches 23659
Number of matches After Lowe's Ratio 3256
Number of Robust matches 2317

 82%|[redacted]  | 50/61 [01:32<00:22,  2.06s/it]

Number of matches 23577
Number of matches After Lowe's Ratio 2903
Number of Robust matches 2227

 84%|[redacted]  | 51/61 [01:33<00:19,  1.95s/it]

Number of matches 20451
Number of matches After Lowe's Ratio 1511
Number of Robust matches 1090

 85%|[redacted]  | 52/61 [01:35<00:16,  1.79s/it]

Number of matches 20682
Number of matches After Lowe's Ratio 1555
Number of Robust matches 1135

 87%|[redacted]  | 53/61 [01:36<00:13,  1.69s/it]

Number of matches 20570
Number of matches After Lowe's Ratio 2220
Number of Robust matches 1445

 89%|[redacted]  | 54/61 [01:38<00:12,  1.76s/it]

Number of matches 24661
Number of matches After Lowe's Ratio 1920
Number of Robust matches 1223

 90%|[redacted]  | 55/61 [01:40<00:10,  1.76s/it]

Number of matches 20801
Number of matches After Lowe's Ratio 2085
Number of Robust matches 1479
```

```
 92%|██████████| 56/61 [01:41<00:08,  1.68s/it]
```

```
Number of matches 21132
Number of matches After Lowe's Ratio 1732
Number of Robust matches 928
```

```
 93%|██████████| 57/61 [01:43<00:06,  1.64s/it]
```

```
Number of matches 23335
Number of matches After Lowe's Ratio 2553
Number of Robust matches 1315
```

```
 95%|██████████| 58/61 [01:45<00:05,  1.70s/it]
```

```
Number of matches 23849
Number of matches After Lowe's Ratio 1718
Number of Robust matches 700
```

```
 97%|██████████| 59/61 [01:47<00:03,  1.75s/it]
```

```
Number of matches 25324
Number of matches After Lowe's Ratio 2556
Number of Robust matches 959
```

```
 98%|██████████| 60/61 [01:49<00:01,  1.83s/it]
  0%|          | 0/40 [00:00<?, ?it/s]
```

```
Number of matches 18401
Number of matches After Lowe's Ratio 779
Number of Robust matches 270
```

```
  2%|          | 1/40 [00:01<00:40,  1.05s/it]
```

```
Number of matches 17218
Number of matches After Lowe's Ratio 870
Number of Robust matches 455
```

```
  5%|          | 2/40 [00:02<00:44,  1.17s/it]
```

```
Number of matches 23487
Number of matches After Lowe's Ratio 1795
Number of Robust matches 1235
```

```
  8%|          | 3/40 [00:04<00:54,  1.47s/it]
```

```
Number of matches 20639
Number of matches After Lowe's Ratio 2112
Number of Robust matches 1327
```

```
 10%|          | 4/40 [00:05<00:52,  1.46s/it]
```

```
Number of matches 20174
Number of matches After Lowe's Ratio 1131
Number of Robust matches 755
```

```
 12%|          | 5/40 [00:07<00:51,  1.47s/it]
```

```
Number of matches 19154
Number of matches After Lowe's Ratio 566
Number of Robust matches 264
```

```
15%|          | 6/40 [00:08<00:47,  1.41s/it]
```

```
Number of matches 16361
Number of matches After Lowe's Ratio 1518
Number of Robust matches 1084
```

```
18%|          | 7/40 [00:09<00:44,  1.36s/it]
```

```
Number of matches 22705
Number of matches After Lowe's Ratio 1067
Number of Robust matches 598
```

```
20%|          | 8/40 [00:11<00:48,  1.53s/it]
```

```
Number of matches 22783
Number of matches After Lowe's Ratio 2500
Number of Robust matches 1855
```

```
22%|          | 9/40 [00:13<00:48,  1.57s/it]
```

```
Number of matches 23505
Number of matches After Lowe's Ratio 2794
Number of Robust matches 1969
```

```
25%|          | 10/40 [00:14<00:48,  1.61s/it]
```

```
Number of matches 20881
Number of matches After Lowe's Ratio 2259
Number of Robust matches 1738
```

```
28%|          | 11/40 [00:16<00:45,  1.58s/it]
```

```
Number of matches 22292
Number of matches After Lowe's Ratio 2275
Number of Robust matches 1409
```

```
30%|          | 12/40 [00:18<00:45,  1.63s/it]
```

```
Number of matches 22440
Number of matches After Lowe's Ratio 1559
Number of Robust matches 1118
```

```
32%|          | 13/40 [00:19<00:44,  1.64s/it]
```

```
Number of matches 23905
Number of matches After Lowe's Ratio 2120
Number of Robust matches 1566
```

```
35%|          | 14/40 [00:21<00:46,  1.77s/it]
```

```
Number of matches 27181
Number of matches After Lowe's Ratio 2046
Number of Robust matches 1321
```

```
Number of Robust matches 1321

 38%|▉▉▉▉      | 15/40 [00:24<00:47,  1.91s/it]

Number of matches 26925
Number of matches After Lowe's Ratio 2388
Number of Robust matches 1353

 40%|▉▉▉▉      | 16/40 [00:26<00:47,  1.96s/it]

Number of matches 28745
Number of matches After Lowe's Ratio 2514
Number of Robust matches 1261

 42%|▉▉▉▉▏     | 17/40 [00:28<00:46,  2.03s/it]

Number of matches 25187
Number of matches After Lowe's Ratio 2381
Number of Robust matches 1252

 45%|▉▉▉▉▉     | 18/40 [00:30<00:46,  2.11s/it]

Number of matches 20447
Number of matches After Lowe's Ratio 1890
Number of Robust matches 869

 48%|▉▉▉▉▉▏    | 19/40 [00:32<00:42,  2.05s/it]

Number of matches 20792
Number of matches After Lowe's Ratio 2102
Number of Robust matches 903

 50%|▉▉▉▉▉▉    | 20/40 [00:34<00:37,  1.88s/it]

Number of matches 20614
Number of matches After Lowe's Ratio 1929
Number of Robust matches 777

 52%|▉▉▉▉▉▉▏   | 21/40 [00:35<00:33,  1.75s/it]

Number of matches 17425
Number of matches After Lowe's Ratio 1492
Number of Robust matches 733

 55%|▉▉▉▉▉▉▉   | 22/40 [00:36<00:28,  1.60s/it]

Number of matches 20156
Number of matches After Lowe's Ratio 1303
Number of Robust matches 733

 57%|▉▉▉▉▉▉▉▏  | 23/40 [00:38<00:27,  1.60s/it]

Number of matches 31550
Number of matches After Lowe's Ratio 520
Number of Robust matches 146

 60%|▉▉▉▉▉▉▉▉  | 24/40 [00:41<00:30,  1.93s/it]
```

```
60%|▌▌▌▌▌▌▌    | 24/40 [00:41<00:30,  1.93s/it]

Number of matches 29090
Number of matches After Lowe's Ratio 1144
Number of Robust matches 461


62%|▌▌▌▌▌▌▌▌    | 25/40 [00:43<00:32,  2.17s/it]

Number of matches 34114
Number of matches After Lowe's Ratio 337
Number of Robust matches 6


65%|▌▌▌▌▌▌▌▌    | 26/40 [00:46<00:32,  2.34s/it]

Number of matches 25453
Number of matches After Lowe's Ratio 943
Number of Robust matches 320


68%|▌▌▌▌▌▌▌▌▌    | 27/40 [00:48<00:29,  2.25s/it]

Number of matches 23227
Number of matches After Lowe's Ratio 1749
Number of Robust matches 778


70%|▌▌▌▌▌▌▌▌▌    | 28/40 [00:50<00:25,  2.09s/it]

Number of matches 21847
Number of matches After Lowe's Ratio 1731
Number of Robust matches 676


72%|▌▌▌▌▌▌▌▌▌▌    | 29/40 [00:51<00:21,  1.92s/it]

Number of matches 18920
Number of matches After Lowe's Ratio 1454
Number of Robust matches 600


75%|▌▌▌▌▌▌▌▌▌▌▌    | 30/40 [00:53<00:17,  1.74s/it]

Number of matches 18788
Number of matches After Lowe's Ratio 1292
Number of Robust matches 454


78%|▌▌▌▌▌▌▌▌▌▌▌▌    | 31/40 [00:54<00:15,  1.68s/it]

Number of matches 18588
Number of matches After Lowe's Ratio 1157
Number of Robust matches 428


80%|▌▌▌▌▌▌▌▌▌▌▌▌▌    | 32/40 [00:56<00:12,  1.61s/it]

Number of matches 19964
Number of matches After Lowe's Ratio 2129
Number of Robust matches 777


82%|▌▌▌▌▌▌▌▌▌▌▌▌▌▌    | 33/40 [00:57<00:10,  1.56s/it]

Number of matches 22044
Number of matches After Lowe's Ratio 1148
```

```
Number of Robust matches 397
```

```
 85%|▉▉▉▉▉▉▉▉▉ | 34/40 [00:59<00:09,  1.57s/it]
```

```
Number of matches 20394
Number of matches After Lowe's Ratio 1955
Number of Robust matches 914
```

```
 88%|▉▉▉▉▉▉▉▉▉▉ | 35/40 [01:00<00:07,  1.54s/it]
```

```
Number of matches 22323
Number of matches After Lowe's Ratio 1621
Number of Robust matches 632
```

```
 90%|▉▉▉▉▉▉▉▉▉▉ | 36/40 [01:02<00:06,  1.71s/it]
```

```
Number of matches 17096
Number of matches After Lowe's Ratio 1273
Number of Robust matches 628
```

```
 92%|▉▉▉▉▉▉▉▉▉▉▉ | 37/40 [01:04<00:04,  1.58s/it]
```

```
Number of matches 17381
Number of matches After Lowe's Ratio 994
Number of Robust matches 655
```

```
 95%|▉▉▉▉▉▉▉▉▉▉▉ | 38/40 [01:05<00:03,  1.54s/it]
```

```
Number of matches 17165
Number of matches After Lowe's Ratio 1323
Number of Robust matches 926
```

```
 98%|▉▉▉▉▉▉▉▉▉▉▉▉| 39/40 [01:06<00:01,  1.71s/it]
```

```
Number of matches 17337
Number of matches After Lowe's Ratio 1321
Number of Robust matches 850
```

In [20]:

```
H_left_orb = []
H_right_orb = []

num_matches_orb = []
num_good_matches_orb = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_orb[j:j+2][::-1],points_all_left_orb[j:j+2][::-1],descriptors_all_left_orb[j:j+2][::-1]
)
    H_left_orb.append(H_a)
    num_matches_orb.append(matches)
    num_good_matches_orb.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
```

```
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_orb[j:j+2][::-1],points_all_right_orb[j:j+2][::-1],descriptors_all_right_orb[j:j+2][:
:-1])
    H_right_orb.append(H_a)
    num_matches_orb.append(matches)
    num_good_matches_orb.append(gd_matches)
```

```
  2%|          | 1/61 [00:00<00:12,  4.98it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 232
Number of Robust matches 46



Number of matches 5000
Number of matches After Lowe's Ratio 256

  5%|          | 3/61 [00:00<00:11,  5.15it/s]

Number of Robust matches 60



Number of matches 5000
Number of matches After Lowe's Ratio 144
Number of Robust matches 7


  8%|          | 5/61 [00:00<00:09,  5.72it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 381
Number of Robust matches 170



Number of matches 5000
Number of matches After Lowe's Ratio 475
Number of Robust matches 257


 11%|          | 7/61 [00:01<00:09,  5.95it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 378
Number of Robust matches 162



Number of matches 5000
Number of matches After Lowe's Ratio 432
Number of Robust matches 257


 15%|          | 9/61 [00:01<00:10,  5.11it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 235
Number of Robust matches 101



Number of matches 5000
Number of matches After Lowe's Ratio 509
Number of Robust matches 374
```

```
 18%|▉         | 11/61 [00:02<00:08,  5.64it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 333
Number of Robust matches 184



Number of matches 5000
Number of matches After Lowe's Ratio 455
Number of Robust matches 307


 20%|▉         | 12/61 [00:02<00:09,  4.95it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 489
Number of Robust matches 338


 23%|▉▏        | 14/61 [00:02<00:10,  4.64it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 471
Number of Robust matches 351



Number of matches 5000
Number of matches After Lowe's Ratio 573
Number of Robust matches 455


 26%|▉▉        | 16/61 [00:03<00:08,  5.27it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 551
Number of Robust matches 409



Number of matches 5000
Number of matches After Lowe's Ratio 603
Number of Robust matches 446


 30%|▉▉▏       | 18/61 [00:03<00:07,  5.69it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 510
Number of Robust matches 361



Number of matches 5000
Number of matches After Lowe's Ratio 565
Number of Robust matches 367


 33%|▉▉▉▏      | 20/61 [00:03<00:06,  5.93it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 667
Number of Robust matches 522



Number of matches 5000
Number of matches After Lowe's Ratio 508
Number of Robust matches 359
```

```
Number of Robust matches 358

 36%|[redacted]       | 22/61 [00:04<00:06,  6.07it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 356
Number of Robust matches 201


Number of matches 5000
Number of matches After Lowe's Ratio 494
Number of Robust matches 338

 39%|[redacted]       | 24/61 [00:04<00:06,  6.10it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 601
Number of Robust matches 401


Number of matches 5000
Number of matches After Lowe's Ratio 468
Number of Robust matches 332

 41%|[redacted]       | 25/61 [00:04<00:06,  5.40it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 460
Number of Robust matches 284

 43%|[redacted]       | 26/61 [00:04<00:07,  4.46it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 414
Number of Robust matches 269

 44%|[redacted]       | 27/61 [00:05<00:07,  4.40it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 490
Number of Robust matches 305

 46%|[redacted]       | 28/61 [00:05<00:08,  4.11it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 382
Number of Robust matches 177

 49%|[redacted]       | 30/61 [00:05<00:06,  4.56it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 298
Number of Robust matches 93


Number of matches 5000
Number of matches After Lowe's Ratio 362
Number of Robust matches 178
```

```
52%|      | 32/61 [00:06<00:05,  5.00it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 274
Number of Robust matches 96


Number of matches 5000
Number of matches After Lowe's Ratio 141
Number of Robust matches 17

56%|      | 34/61 [00:06<00:04,  5.58it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 488
Number of Robust matches 242


Number of matches 5000
Number of matches After Lowe's Ratio 533
Number of Robust matches 290

59%|      | 36/61 [00:06<00:04,  5.84it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 394
Number of Robust matches 199


Number of matches 5000
Number of matches After Lowe's Ratio 422
Number of Robust matches 233

62%|      | 38/61 [00:07<00:03,  5.96it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 330
Number of Robust matches 140


Number of matches 5000
Number of matches After Lowe's Ratio 363
Number of Robust matches 126

66%|      | 40/61 [00:07<00:03,  6.04it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 279
Number of Robust matches 114


Number of matches 5000
Number of matches After Lowe's Ratio 453
Number of Robust matches 295

69%|      | 42/61 [00:07<00:03,  6.10it/s]
```

```
Number of matches 5000
Number of matches After Lowe's Ratio 528
Number of Robust matches 352



Number of matches 5000
Number of matches After Lowe's Ratio 624
Number of Robust matches 469


 72%|[redacted]  | 44/61 [00:08<00:02,  6.04it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 685
Number of Robust matches 529



Number of matches 5000
Number of matches After Lowe's Ratio 584
Number of Robust matches 404


 75%|[redacted]  | 46/61 [00:08<00:02,  6.15it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 505
Number of Robust matches 333



Number of matches 5000
Number of matches After Lowe's Ratio 467
Number of Robust matches 262


 79%|[redacted]  | 48/61 [00:08<00:02,  6.17it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 307
Number of Robust matches 133



Number of matches 5000
Number of matches After Lowe's Ratio 289
Number of Robust matches 166


 82%|[redacted]  | 50/61 [00:09<00:01,  6.17it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 643
Number of Robust matches 495



Number of matches 5000
Number of matches After Lowe's Ratio 654
Number of Robust matches 516


 85%|[redacted]  | 52/61 [00:09<00:01,  5.27it/s]
Number of matches 5000
Number of matches After Lowe's Ratio 447
Number of Robust matches 304
```

```
Number of matches 5000
Number of matches After Lowe's Ratio 377
Number of Robust matches 209


 89%|[redacted] | 54/61 [00:09<00:01,  5.38it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 605
Number of Robust matches 428


Number of matches 5000
Number of matches After Lowe's Ratio 350
Number of Robust matches 221


 92%|[redacted] | 56/61 [00:10<00:00,  5.68it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 512
Number of Robust matches 336


Number of matches 5000
Number of matches After Lowe's Ratio 409
Number of Robust matches 221


 95%|[redacted] | 58/61 [00:10<00:00,  5.82it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 479
Number of Robust matches 253


Number of matches 5000
Number of matches After Lowe's Ratio 316
Number of Robust matches 119


 97%|[redacted] | 59/61 [00:10<00:00,  5.91it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 438
Number of Robust matches 166


Number of matches 5000
Number of matches After Lowe's Ratio 217

 98%|[redacted] | 60/61 [00:10<00:00,  5.46it/s]
  2%|[redacted]          | 1/40 [00:00<00:06,  6.16it/s]

Number of Robust matches 25


Number of matches 5000
Number of matches After Lowe's Ratio 319
Number of Robust matches 161
```

```
  8%|▌         | 3/40 [00:00<00:05,  6.18it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 588
Number of Robust matches 439



Number of matches 5000
Number of matches After Lowe's Ratio 497
Number of Robust matches 377


 12%|▌         | 5/40 [00:00<00:05,  6.13it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 359
Number of Robust matches 213



Number of matches 5000
Number of matches After Lowe's Ratio 190
Number of Robust matches 76


 18%|▌         | 7/40 [00:01<00:05,  6.11it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 480
Number of Robust matches 317



Number of matches 5000
Number of matches After Lowe's Ratio 292
Number of Robust matches 135


 22%|▌         | 9/40 [00:01<00:05,  6.16it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 678
Number of Robust matches 546



Number of matches 5000
Number of matches After Lowe's Ratio 785
Number of Robust matches 603


 28%|▌         | 11/40 [00:01<00:04,  6.14it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 551
Number of Robust matches 341



Number of matches 5000
Number of matches After Lowe's Ratio 621
Number of Robust matches 431


 32%|▌         | 13/40 [00:02<00:04,  6.03it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 400
```

```
Number of Robust matches 246



Number of matches 5000
Number of matches After Lowe's Ratio 572
Number of Robust matches 397


 35%|[redacted]       | 14/40 [00:02<00:04,  5.24it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 565
Number of Robust matches 378


 40%|[redacted]       | 16/40 [00:02<00:05,  4.78it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 556
Number of Robust matches 378



Number of matches 5000
Number of matches After Lowe's Ratio 533
Number of Robust matches 346


 45%|[redacted]       | 18/40 [00:03<00:04,  5.41it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 443
Number of Robust matches 282



Number of matches 5000
Number of matches After Lowe's Ratio 436
Number of Robust matches 261


 50%|[redacted]       | 20/40 [00:03<00:03,  5.78it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 544
Number of Robust matches 286



Number of matches 5000
Number of matches After Lowe's Ratio 565
Number of Robust matches 299


 55%|[redacted]       | 22/40 [00:03<00:03,  5.97it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 542
Number of Robust matches 289



Number of matches 5000
Number of matches After Lowe's Ratio 552
Number of Robust matches 398
```

```
 57%|[redacted]     | 23/40 [00:04<00:02,  5.70it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 181
Number of Robust matches 38



Number of matches 5000
Number of matches After Lowe's Ratio 224
 62%|[redacted]     | 25/40 [00:04<00:02,  5.57it/s]

Number of Robust matches 70



Number of matches 5000
Number of matches After Lowe's Ratio 137
Number of Robust matches 7


 68%|[redacted]     | 27/40 [00:04<00:02,  5.84it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 247
Number of Robust matches 83



Number of matches 5000
Number of matches After Lowe's Ratio 439
Number of Robust matches 189


 72%|[redacted]     | 29/40 [00:05<00:01,  5.74it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 410
Number of Robust matches 186



Number of matches 5000
Number of matches After Lowe's Ratio 453
Number of Robust matches 217


 78%|[redacted]     | 31/40 [00:05<00:01,  5.71it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 343
Number of Robust matches 113



Number of matches 5000
Number of matches After Lowe's Ratio 349
Number of Robust matches 104


 82%|[redacted]     | 33/40 [00:05<00:01,  5.81it/s]

Number of matches 5000
Number of matches After Lowe's Ratio 549
Number of Robust matches 262



Number of matches 5000
Number of matches After Lowe's Ratio 200
```

```
Number of matches After Lowe's Ratio 299
Number of Robust matches 112
```

```
 88%|██████████| 35/40 [00:06<00:00,  5.29it/s]
```

```
Number of matches 5000
Number of matches After Lowe's Ratio 448
Number of Robust matches 255



Number of matches 5000
Number of matches After Lowe's Ratio 385
Number of Robust matches 188
```

```
 92%|██████████| 37/40 [00:06<00:00,  5.70it/s]
```

```
Number of matches 5000
Number of matches After Lowe's Ratio 345
Number of Robust matches 150



Number of matches 5000
Number of matches After Lowe's Ratio 356
Number of Robust matches 239
```

```
 98%|██████████| 39/40 [00:06<00:00,  5.71it/s]
```

```
Number of matches 5000
Number of matches After Lowe's Ratio 692
Number of Robust matches 584



Number of matches 5000
Number of matches After Lowe's Ratio 584
Number of Robust matches 437
```

In [ ]:

```
H_left_akaze = []
H_right_akaze = []

num_matches_akaze = []
num_good_matches_akaze = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_akaze[j:j+2][::-1],points_all_left_akaze[j:j+2][::-1],descriptors_all_left_akaze[j:j+2]
[::-1])
    H_left_akaze.append(H_a)
    num_matches_akaze.append(matches)
    num_good_matches_akaze.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_akaze[j:j+2][::-1],points_all_right_akaze[j:j+2][::-1],descriptors_all_right_akaze[j:
```

```
j+2][::-1])
    H_right_akaze.append(H_a)
    num_matches_akaze.append(matches)
    num_good_matches_akaze.append(gd_matches)
```

In [ ]:

```
H_left_kaze = []
H_right_kaze = []

num_matches_kaze = []
num_good_matches_kaze = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_kaze[j:j+2][::-1],points_all_left_kaze[j:j+2][::-1],descriptors_all_left_kaze[j:j+2][::
-1])
    H_left_kaze.append(H_a)
    num_matches_kaze.append(matches)
    num_good_matches_kaze.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_kaze[j:j+2][::-1],points_all_right_kaze[j:j+2][::-1],descriptors_all_right_kaze[j:j+2
][::-1])
    H_right_kaze.append(H_a)
    num_matches_kaze.append(matches)
    num_good_matches_kaze.append(gd_matches)
```

In [ ]:

```
H_left_freak = []
H_right_freak = []

num_matches_freak = []
num_good_matches_freak = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_freak[j:j+2][::-1],points_all_left_freak[j:j+2][::-1],descriptors_all_left_freak[j:j+2]
[::-1])
    H_left_freak.append(H_a)
    num_matches_freak.append(matches)
    num_good_matches_freak.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_freak[j:j+2][::-1],points_all_right_freak[j:j+2][::-1],descriptors_all_right_freak[j:
j+2][::-1])
    H_right_freak.append(H_a)
    num_matches_freak.append(matches)
    num_good_matches_freak.append(gd_matches)
```

In [ ]:

```
H_left_mser = []
H_right_mser = []

num_matches_mser = []
```

```
num_good_matches_mser = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_mser[j:j+2][::-1],points_all_left_mser[j:j+2][::-1],descriptors_all_left_mser[j:j+2][::
-1])
    H_left_mser.append(H_a)
    num_matches_mser.append(matches)
    num_good_matches_mser.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_mser[j:j+2][::-1],points_all_right_mser[j:j+2][::-1],descriptors_all_right_mser[j:j+2
][::-1])
    H_right_mser.append(H_a)
    num_matches_mser.append(matches)
    num_good_matches_mser.append(gd_matches)
```

In [ ]:

```
H_left_superpoint = []
H_right_superpoint = []

num_matches_superpoint = []
num_good_matches_superpoint = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoint_all_left_
superpoint[j:j+2][::-1],point_all_left_superpoint[j:j+2][::-1],descriptor_all_left_super
point[j:j+2][::-1])
    H_left_superpoint.append(H_a)
    num_matches_superpoint.append(matches)
    num_good_matches_superpoint.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_superpoint[j:j+2][::-1],points_all_right_superpoint[j:j+2][::-1],descriptors_all_righ
t_superpoint[j:j+2][::-1])
    H_right_superpoint.append(H_a)
    num_matches_superpoint.append(matches)
    num_good_matches_superpoint.append(gd_matches)
```

In [ ]:

```
H_left_gftt = []
H_right_gftt = []

num_matches_gftt = []
num_good_matches_gftt = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_gftt[j:j+2][::-1],points_all_left_gftt[j:j+2][::-1],descriptors_all_left_gftt[j:j+2][::
-1])
    H_left_gftt.append(H_a)
    num_matches_gftt.append(matches)
```

```
    num_good_matches_gftt.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_gftt[j:j+2][::-1],points_all_right_gftt[j:j+2][::-1],descriptors_all_right_gftt[j:j+2
][::-1])
    H_right_gftt.append(H_a)
    num_matches_gftt.append(matches)
    num_good_matches_gftt.append(gd_matches)
```

In [ ]:

```
H_left_daisy = []
H_right_daisy = []

num_matches_daisy = []
num_good_matches_daisy = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_daisy[j:j+2][::-1],points_all_left_daisy[j:j+2][::-1],descriptors_all_left_daisy[j:j+2]
[::-1])
    H_left_daisy.append(H_a)
    num_matches_daisy.append(matches)
    num_good_matches_daisy.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_daisy[j:j+2][::-1],points_all_right_daisy[j:j+2][::-1],descriptors_all_right_daisy[j:
j+2][::-1])
    H_right_daisy.append(H_a)
    num_matches_daisy.append(matches)
    num_good_matches_daisy.append(gd_matches)
```

In [22]:

```
H_left_fast = []
H_right_fast = []

num_matches_fast = []
num_good_matches_fast = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_fast[j:j+2][::-1],points_all_left_fast[j:j+2][::-1],descriptors_all_left_fast[j:j+2][::
-1])
    H_left_fast.append(H_a)
    num_matches_fast.append(matches)
    num_good_matches_fast.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_fast[j:j+2][::-1],points_all_right_fast[j:j+2][::-1],descriptors_all_right_fast[j:j+2
][::-1])
    H_right_fast.append(H_a)
    num_matches_fast.append(matches)
```

```
    num_good_matches_fast.append(gd_matches)
```

```
  2%|          | 1/61 [00:12<12:29, 12.49s/it]
```

```
Number of matches 68911
Number of matches After Lowe's Ratio 3300
Number of Robust matches 1298
```

```
  3%|          | 2/61 [00:25<12:52, 13.09s/it]
```

```
Number of matches 76210
Number of matches After Lowe's Ratio 1361
Number of Robust matches 518
```

```
  5%|          | 3/61 [00:40<13:14, 13.71s/it]
```

```
Number of matches 66951
Number of matches After Lowe's Ratio 83
Number of Robust matches 16
```

```
  7%|          | 4/61 [00:52<12:34, 13.24s/it]
```

```
Number of matches 65766
Number of matches After Lowe's Ratio 11047
Number of Robust matches 5331
```

```
  8%|          | 5/61 [01:05<12:04, 12.94s/it]
```

```
Number of matches 69653
Number of matches After Lowe's Ratio 747
Number of Robust matches 276
```

```
 10%|          | 6/61 [01:18<11:59, 13.08s/it]
```

```
Number of matches 65636
Number of matches After Lowe's Ratio 3203
Number of Robust matches 1454
```

```
 11%|          | 7/61 [01:32<11:50, 13.16s/it]
```

```
Number of matches 73090
Number of matches After Lowe's Ratio 1397
Number of Robust matches 808
```

```
 13%|          | 8/61 [01:45<11:37, 13.16s/it]
```

```
Number of matches 52923
Number of matches After Lowe's Ratio 4817
Number of Robust matches 2746
```

```
 15%|          | 9/61 [01:56<10:48, 12.47s/it]
```

```
Number of matches 65649
Number of matches After Lowe's Ratio 4888
Number of Robust matches 3359
```

```
 16%|          | 10/61 [02:07<10:20, 12.16s/it]
```

```
Number of matches 40883
Number of matches After Lowe's Ratio 1797
Number of Robust matches 1113


 18%|☃         | 11/61 [02:15<09:02, 10.86s/it]

Number of matches 56500
Number of matches After Lowe's Ratio 7939
Number of Robust matches 6024


 20%|☃         | 12/61 [02:26<08:55, 10.92s/it]

Number of matches 50020
Number of matches After Lowe's Ratio 522
Number of Robust matches 342


 21%|☃         | 13/61 [02:36<08:31, 10.65s/it]

Number of matches 62854
Number of matches After Lowe's Ratio 2605
Number of Robust matches 1852


 23%|☃         | 14/61 [02:49<08:47, 11.22s/it]

Number of matches 63693
Number of matches After Lowe's Ratio 5901
Number of Robust matches 4347


 25%|☃         | 15/61 [03:02<09:00, 11.75s/it]

Number of matches 61955
Number of matches After Lowe's Ratio 7981
Number of Robust matches 5963


 26%|☃         | 16/61 [03:14<08:56, 11.92s/it]

Number of matches 61237
Number of matches After Lowe's Ratio 5495
Number of Robust matches 3881


 28%|☃         | 17/61 [03:27<08:53, 12.13s/it]

Number of matches 65573
Number of matches After Lowe's Ratio 9202
Number of Robust matches 7421


 30%|☃         | 18/61 [03:40<08:57, 12.51s/it]

Number of matches 68095
Number of matches After Lowe's Ratio 18645
Number of Robust matches 13747


 31%|☃         | 19/61 [03:54<09:01, 12.89s/it]

Number of matches 68833
Number of matches After Lowe's Ratio 17013
Number of Robust matches 14069
```

```
 33%|[redacted]       | 20/61 [04:08<09:07, 13.35s/it]

Number of matches 74743
Number of matches After Lowe's Ratio 10602
Number of Robust matches 6604


 34%|[redacted]       | 21/61 [04:23<09:14, 13.86s/it]

Number of matches 74966
Number of matches After Lowe's Ratio 3094
Number of Robust matches 1865


 36%|[redacted]       | 22/61 [04:39<09:17, 14.29s/it]

Number of matches 70972
Number of matches After Lowe's Ratio 18571
Number of Robust matches 13777


Number of matches 71729
Number of matches After Lowe's Ratio 3443
 38%|[redacted]       | 23/61 [04:53<09:02, 14.28s/it]

Number of Robust matches 2001


 39%|[redacted]       | 24/61 [05:07<08:49, 14.32s/it]

Number of matches 74796
Number of matches After Lowe's Ratio 14006
Number of Robust matches 9423


 41%|[redacted]       | 25/61 [05:22<08:42, 14.50s/it]

Number of matches 82185
Number of matches After Lowe's Ratio 69
Number of Robust matches 13


 43%|[redacted]       | 26/61 [05:38<08:43, 14.97s/it]

Number of matches 72748
Number of matches After Lowe's Ratio 337
Number of Robust matches 105


 44%|[redacted]       | 27/61 [05:52<08:20, 14.73s/it]

Number of matches 73262
Number of matches After Lowe's Ratio 9567
Number of Robust matches 5749


 46%|[redacted]       | 28/61 [06:07<08:00, 14.57s/it]

Number of matches 71040
Number of matches After Lowe's Ratio 289
Number of Robust matches 111


 48%|[redacted]       | 29/61 [06:21<07:42, 14.48s/it]
```

```
48%|[redacted]| 29/61 [06:21<07:43, 14.48s/it]

Number of matches 76828
Number of matches After Lowe's Ratio 203
Number of Robust matches 80


49%|[redacted]| 30/61 [06:36<07:34, 14.66s/it]

Number of matches 78791
Number of matches After Lowe's Ratio 5045
Number of Robust matches 2314


51%|[redacted]| 31/61 [06:51<07:26, 14.88s/it]

Number of matches 73679
Number of matches After Lowe's Ratio 3179
Number of Robust matches 1679


52%|[redacted]| 32/61 [07:06<07:06, 14.72s/it]

Number of matches 67557
Number of matches After Lowe's Ratio 38
Number of Robust matches 9


54%|[redacted]| 33/61 [07:19<06:43, 14.40s/it]

Number of matches 68685
Number of matches After Lowe's Ratio 13637
Number of Robust matches 9119


56%|[redacted]| 34/61 [07:32<06:16, 13.93s/it]

Number of matches 65206
Number of matches After Lowe's Ratio 13088
Number of Robust matches 8389


57%|[redacted]| 35/61 [07:45<05:54, 13.62s/it]

Number of matches 71629
Number of matches After Lowe's Ratio 11553
Number of Robust matches 6689


59%|[redacted]| 36/61 [08:00<05:49, 13.97s/it]

Number of matches 78286
Number of matches After Lowe's Ratio 10592
Number of Robust matches 6332


61%|[redacted]| 37/61 [08:15<05:45, 14.41s/it]

Number of matches 86349
Number of matches After Lowe's Ratio 13402
Number of Robust matches 7436


62%|[redacted]| 38/61 [08:33<05:52, 15.32s/it]

Number of matches 92020
Number of matches After Lowe's Ratio 16179
```

```
Number of Robust matches 8053
```

```
 64%|▊▊▊▊▊▊▊    | 39/61 [08:50<05:49, 15.90s/it]
```

```
Number of matches 86414
Number of matches After Lowe's Ratio 15117
Number of Robust matches 8060
```

```
 66%|▊▊▊▊▊▊▊    | 40/61 [09:07<05:38, 16.14s/it]
```

```
Number of matches 75915
Number of matches After Lowe's Ratio 16320
Number of Robust matches 9075
```

```
 67%|▊▊▊▊▊▊▊    | 41/61 [09:21<05:14, 15.70s/it]
```

```
Number of matches 71589
Number of matches After Lowe's Ratio 18361
Number of Robust matches 13911
```

```
 69%|▊▊▊▊▊▊▊    | 42/61 [09:36<04:49, 15.25s/it]
```

```
Number of matches 67342
Number of matches After Lowe's Ratio 18395
Number of Robust matches 13882
```

```
 70%|▊▊▊▊▊▊▊    | 43/61 [09:49<04:23, 14.63s/it]
```

```
Number of matches 65621
Number of matches After Lowe's Ratio 16144
Number of Robust matches 12050
```

```
 72%|▊▊▊▊▊▊▊    | 44/61 [10:02<04:03, 14.31s/it]
```

```
Number of matches 70168
Number of matches After Lowe's Ratio 15654
Number of Robust matches 11060
```

```
 74%|▊▊▊▊▊▊▊    | 45/61 [10:17<03:48, 14.29s/it]
```

```
Number of matches 72917
Number of matches After Lowe's Ratio 16564
Number of Robust matches 11721
```

```
 75%|▊▊▊▊▊▊▊    | 46/61 [10:31<03:35, 14.36s/it]
```

```
Number of matches 70010
Number of matches After Lowe's Ratio 18575
Number of Robust matches 12141
```

```
 77%|▊▊▊▊▊▊▊    | 47/61 [10:46<03:22, 14.45s/it]
```

```
Number of matches 67372
Number of matches After Lowe's Ratio 13509
Number of Robust matches 8806
```

```
 79%|▉▉▉▉▉▉▉▉  | 48/61 [10:59<03:02, 14.01s/it]

Number of matches 57035
Number of matches After Lowe's Ratio 7445
Number of Robust matches 4726


 80%|▉▉▉▉▉▉▉▉  | 49/61 [11:10<02:39, 13.25s/it]

Number of matches 53539
Number of matches After Lowe's Ratio 16844
Number of Robust matches 13740


 82%|▉▉▉▉▉▉▉▉  | 50/61 [11:21<02:17, 12.52s/it]

Number of matches 57411
Number of matches After Lowe's Ratio 13289
Number of Robust matches 11307


 84%|▉▉▉▉▉▉▉▉  | 51/61 [11:32<02:00, 12.06s/it]

Number of matches 57710
Number of matches After Lowe's Ratio 12598
Number of Robust matches 8245


 85%|▉▉▉▉▉▉▉▉  | 52/61 [11:43<01:46, 11.79s/it]

Number of matches 55630
Number of matches After Lowe's Ratio 12001
Number of Robust matches 9153


 87%|▉▉▉▉▉▉▉▉  | 53/61 [11:54<01:32, 11.51s/it]

Number of matches 58740
Number of matches After Lowe's Ratio 18109
Number of Robust matches 14293


 89%|▉▉▉▉▉▉▉▉  | 54/61 [12:05<01:19, 11.42s/it]

Number of matches 58101
Number of matches After Lowe's Ratio 5612
Number of Robust matches 3761


 90%|▉▉▉▉▉▉▉▉  | 55/61 [12:17<01:08, 11.46s/it]

Number of matches 54928
Number of matches After Lowe's Ratio 3708
Number of Robust matches 2251


 92%|▉▉▉▉▉▉▉▉  | 56/61 [12:27<00:56, 11.23s/it]

Number of matches 59193
Number of matches After Lowe's Ratio 8403
Number of Robust matches 4716


 93%|▉▉▉▉▉▉▉▉  | 57/61 [12:39<00:45, 11.40s/it]

Number of matches 62057
Number of matches After Lowe's Ratio 7232
```

```
Number of Robust matches 3884


 95%|██████████| 58/61 [12:52<00:35, 11.78s/it]

Number of matches 61730
Number of matches After Lowe's Ratio 4388
Number of Robust matches 1899


 97%|██████████| 59/61 [13:04<00:23, 11.98s/it]

Number of matches 63507
Number of matches After Lowe's Ratio 8205
Number of Robust matches 4418


 98%|██████████| 60/61 [13:16<00:13, 13.28s/it]
  0%|          | 0/40 [00:00<?, ?it/s]

Number of matches 56369
Number of matches After Lowe's Ratio 782
Number of Robust matches 320


  2%|▏         | 1/40 [00:12<08:16, 12.74s/it]

Number of matches 68332
Number of matches After Lowe's Ratio 4843
Number of Robust matches 2770


  5%|▌         | 2/40 [00:26<08:25, 13.30s/it]

Number of matches 79794
Number of matches After Lowe's Ratio 14527
Number of Robust matches 10346


  8%|▉         | 3/40 [00:41<08:45, 14.21s/it]

Number of matches 63717
Number of matches After Lowe's Ratio 12785
Number of Robust matches 9009


 10%|▉         | 4/40 [00:52<07:44, 12.91s/it]

Number of matches 36944
Number of matches After Lowe's Ratio 4621
Number of Robust matches 3388


 12%|▉         | 5/40 [00:59<06:20, 10.87s/it]

Number of matches 46830
Number of matches After Lowe's Ratio 2727
Number of Robust matches 1828


 15%|▉         | 6/40 [01:08<05:46, 10.20s/it]

Number of matches 36288
Number of matches After Lowe's Ratio 7325
Number of Robust matches 5390
```

```
 18%|          | 7/40 [01:16<05:09,  9.37s/it]

Number of matches 66719
Number of matches After Lowe's Ratio 6507
Number of Robust matches 4500


 20%|          | 8/40 [01:29<05:41, 10.67s/it]

Number of matches 67675
Number of matches After Lowe's Ratio 14965
Number of Robust matches 12625


 22%|          | 9/40 [01:44<06:04, 11.75s/it]

Number of matches 70144
Number of matches After Lowe's Ratio 14774
Number of Robust matches 11706


 25%|          | 10/40 [01:57<06:11, 12.40s/it]

Number of matches 68163
Number of matches After Lowe's Ratio 17861
Number of Robust matches 14861


Number of matches 77245
Number of matches After Lowe's Ratio 24380
 28%|          | 11/40 [02:12<06:20, 13.14s/it]

Number of Robust matches 18118


 30%|          | 12/40 [02:28<06:28, 13.88s/it]

Number of matches 80940
Number of matches After Lowe's Ratio 7144
Number of Robust matches 5052


 32%|          | 13/40 [02:44<06:33, 14.58s/it]

Number of matches 83041
Number of matches After Lowe's Ratio 12144
Number of Robust matches 8702


Number of matches 86229
Number of matches After Lowe's Ratio 14005
 35%|          | 14/40 [03:01<06:39, 15.36s/it]

Number of Robust matches 10483


Number of matches 84754
Number of matches After Lowe's Ratio 10778
Number of Robust matches 6427


 40%|          | 16/40 [03:34<06:23, 16.00s/it]
```

```
Number of matches 82445
Number of matches After Lowe's Ratio 6739
Number of Robust matches 3958


 42%|[redacted]| 17/40 [03:50<06:06, 15.92s/it]

Number of matches 76231
Number of matches After Lowe's Ratio 13840
Number of Robust matches 7856


 45%|[redacted]| 18/40 [04:05<05:43, 15.61s/it]

Number of matches 70850
Number of matches After Lowe's Ratio 11254
Number of Robust matches 5323


 48%|[redacted]| 19/40 [04:19<05:17, 15.14s/it]

Number of matches 71170
Number of matches After Lowe's Ratio 15950
Number of Robust matches 8142


 50%|[redacted]| 20/40 [04:34<04:58, 14.95s/it]

Number of matches 68586
Number of matches After Lowe's Ratio 16123
Number of Robust matches 8179


 52%|[redacted]| 21/40 [04:47<04:36, 14.53s/it]

Number of matches 63817
Number of matches After Lowe's Ratio 4785
Number of Robust matches 2712


 55%|[redacted]| 22/40 [05:00<04:14, 14.15s/it]

Number of matches 73224
Number of matches After Lowe's Ratio 5781
Number of Robust matches 3253


 57%|[redacted]| 23/40 [05:15<04:04, 14.40s/it]

Number of matches 92358
Number of matches After Lowe's Ratio 2296
Number of Robust matches 1140


Number of matches 81708
Number of matches After Lowe's Ratio 9884

 60%|[redacted]| 24/40 [05:34<04:08, 15.54s/it]

Number of Robust matches 4522


 62%|[redacted]| 25/40 [05:50<03:58, 15.87s/it]

Number of matches 96747
Number of matches After Lowe's Ratio 29
Number of Robust matches 7
```

```
Number of Robust matches 7

 65%|▊▊▊▊▊▊▊   | 26/40 [06:09<03:53, 16.70s/it]

Number of matches 80559
Number of matches After Lowe's Ratio 7539
Number of Robust matches 3267

 68%|▊▊▊▊▊▊▊▊  | 27/40 [06:24<03:32, 16.33s/it]

Number of matches 78227
Number of matches After Lowe's Ratio 5022
Number of Robust matches 2078

 70%|▊▊▊▊▊▊▊▊  | 28/40 [06:40<03:12, 16.08s/it]

Number of matches 68755
Number of matches After Lowe's Ratio 12753
Number of Robust matches 6186

 72%|▊▊▊▊▊▊▊▊▌ | 29/40 [06:53<02:48, 15.29s/it]

Number of matches 66026
Number of matches After Lowe's Ratio 4008
Number of Robust matches 2114

 75%|▊▊▊▊▊▊▊▊▊ | 30/40 [07:06<02:24, 14.42s/it]

Number of matches 58240
Number of matches After Lowe's Ratio 7516
Number of Robust matches 3245

 78%|▊▊▊▊▊▊▊▊▊▌ | 31/40 [07:18<02:02, 13.65s/it]

Number of matches 61978
Number of matches After Lowe's Ratio 11839
Number of Robust matches 5352

 80%|▊▊▊▊▊▊▊▊▊▊ | 32/40 [07:30<01:47, 13.39s/it]

Number of matches 64642
Number of matches After Lowe's Ratio 19537
Number of Robust matches 9353

 82%|▊▊▊▊▊▊▊▊▊▌ | 33/40 [07:44<01:34, 13.52s/it]

Number of matches 69416
Number of matches After Lowe's Ratio 9646
Number of Robust matches 5008

 85%|▊▊▊▊▊▊▊▊▊▊▌ | 34/40 [07:58<01:21, 13.50s/it]

Number of matches 65299
Number of matches After Lowe's Ratio 396
Number of Robust matches 157
```

```
Number of matches 68992
Number of matches After Lowe's Ratio 7910
```

```
 88%|[redacted] | 35/40 [08:11<01:06, 13.39s/it]
```

```
Number of Robust matches 5033
```

```
 90%|[redacted] | 36/40 [08:24<00:53, 13.48s/it]
```

```
Number of matches 68202
Number of matches After Lowe's Ratio 5873
Number of Robust matches 3643
```

```
 92%|[redacted] | 37/40 [08:38<00:40, 13.50s/it]
```

```
Number of matches 71476
Number of matches After Lowe's Ratio 3691
Number of Robust matches 2502
```

```
 95%|[redacted] | 38/40 [08:53<00:27, 13.85s/it]
```

```
Number of matches 73450
Number of matches After Lowe's Ratio 7596
Number of Robust matches 5538
```

```
 98%|[redacted] | 39/40 [09:07<00:14, 14.03s/it]
```

```
Number of matches 67118
Number of matches After Lowe's Ratio 5915
Number of Robust matches 4593
```

In [ ]:

```
H_left_star = []
H_right_star = []

num_matches_star = []
num_good_matches_star = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_star[j:j+2][::-1],points_all_left_star[j:j+2][::-1],descriptors_all_left_brief[j:j+2][:
:-1])
    H_left_star.append(H_a)
    num_matches_star.append(matches)
    num_good_matches_star.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_star[j:j+2][::-1],points_all_right_star[j:j+2][::-1],descriptors_all_right_brief[j:j+
2][::-1])
    H_right_star.append(H_a)
    num_matches_star.append(matches)
    num_good_matches_star.append(gd_matches)
```

In [ ]:

```
H_left_sift = []
H_right_sift = []

num_matches_sift = []
num_good_matches_sift = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_sift[j:j+2][::-1],points_all_left_sift[j:j+2][::-1],descriptors_all_left_sift[j:j+2][::
-1])
    H_left_sift.append(H_a)
    num_matches_sift.append(matches)
    num_good_matches_sift.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_sift[j:j+2][::-1],points_all_right_sift[j:j+2][::-1],descriptors_all_right_sift[j:j+2
][::-1])
    H_right_sift.append(H_a)
    num_matches_sift.append(matches)
    num_good_matches_sift.append(gd_matches)
```

In [ ]:

```
H_left_surf = []
H_right_surf = []

num_matches_surf = []
num_good_matches_surf = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_surf[j:j+2][::-1],points_all_left_surf[j:j++2][::-1],descriptors_all_left_surf[j:j+2][:
:-1])
    H_left_surf.append(H_a)
    num_matches_surf.append(matches)
    num_good_matches_surf.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_surf[j:j+2][::-1],points_all_right_surf[j:j+2][::-1],descriptors_all_right_surf[j:j+2
][::-1])
    H_right_surf.append(H_a)
    num_matches_surf.append(matches)
    num_good_matches_surf.append(gd_matches)
```

In [23]:

```
H_left_surfsift = []
H_right_surfsift = []

num_matches_surfsift = []
num_good_matches_surfsift = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
```

```
_surfsift[j:j+2][::-1],points_all_left_surfsift[j:j++2][::-1],descriptors_all_left_surfs
ift[j:j+2][::-1])
    H_left_surfsift.append(H_a)
    num_matches_surfsift.append(matches)
    num_good_matches_surfsift.append(gd_matches)

for j in tqdm(range(len(images_right))):
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_surfsift[j:j+2][::-1],points_all_right_surfsift[j:j+2][::-1],descriptors_all_right_su
rfsift[j:j+2][::-1])
    H_right_surfsift.append(H_a)
    num_matches_surfsift.append(matches)
    num_good_matches_surfsift.append(gd_matches)
```

```
  2%|          | 1/61 [00:04<04:37,  4.63s/it]

Number of matches 26639
Number of matches After Lowe's Ratio 2999
Number of Robust matches 1033


  3%|          | 2/61 [00:09<04:25,  4.50s/it]

Number of matches 28833
Number of matches After Lowe's Ratio 2729
Number of Robust matches 836


  5%|          | 3/61 [00:13<04:27,  4.62s/it]

Number of matches 25864
Number of matches After Lowe's Ratio 580
Number of Robust matches 108


  7%|          | 4/61 [00:18<04:22,  4.61s/it]

Number of matches 27262
Number of matches After Lowe's Ratio 5826
Number of Robust matches 2360


  8%|          | 5/61 [00:22<04:13,  4.53s/it]

Number of matches 25941
Number of matches After Lowe's Ratio 5504
Number of Robust matches 2000


 10%|          | 6/61 [00:27<04:05,  4.47s/it]

Number of matches 25482
Number of matches After Lowe's Ratio 4996
Number of Robust matches 1961


 11%|          | 7/61 [00:31<03:53,  4.32s/it]

Number of matches 27711
Number of matches After Lowe's Ratio 5548
Number of Robust matches 2044


 13%|          | 8/61 [00:35<03:56,  4.46s/it]
```

```
Number of matches 22634
Number of matches After Lowe's Ratio 2857
Number of Robust matches 1255
```

```
 15%|          | 9/61 [00:39<03:39,  4.22s/it]
```

```
Number of matches 27277
Number of matches After Lowe's Ratio 4854
Number of Robust matches 2143
```

```
 16%|          | 10/61 [00:43<03:33,  4.18s/it]
```

```
Number of matches 21475
Number of matches After Lowe's Ratio 2434
Number of Robust matches 888
```

```
 18%|          | 11/61 [00:47<03:20,  4.02s/it]
```

```
Number of matches 25226
Number of matches After Lowe's Ratio 5587
Number of Robust matches 2640
```

```
 20%|          | 12/61 [00:51<03:24,  4.17s/it]
```

```
Number of matches 24931
Number of matches After Lowe's Ratio 5492
Number of Robust matches 2533
```

```
 21%|          | 13/61 [00:55<03:17,  4.12s/it]
```

```
Number of matches 27325
Number of matches After Lowe's Ratio 5220
Number of Robust matches 2344
```

```
 23%|          | 14/61 [01:00<03:23,  4.33s/it]
```

```
Number of matches 29153
Number of matches After Lowe's Ratio 8792
Number of Robust matches 4811
```

```
 25%|          | 15/61 [01:05<03:26,  4.49s/it]
```

```
Number of matches 27669
Number of matches After Lowe's Ratio 6583
Number of Robust matches 3582
```

```
 26%|          | 16/61 [01:10<03:26,  4.58s/it]
```

```
Number of matches 27878
Number of matches After Lowe's Ratio 7099
Number of Robust matches 4201
```

```
 28%|          | 17/61 [01:14<03:21,  4.57s/it]
```

```
Number of matches 27069
Number of matches After Lowe's Ratio 7020
Number of Robust matches 4506
```

```
 30%|[redacted]        | 18/61 [01:19<03:19,  4.63s/it]
Number of matches 26702
Number of matches After Lowe's Ratio 7545
Number of Robust matches 4428

 31%|[redacted]        | 19/61 [01:24<03:15,  4.65s/it]
Number of matches 26111
Number of matches After Lowe's Ratio 8507
Number of Robust matches 4437

 33%|[redacted]        | 20/61 [01:28<03:04,  4.49s/it]
Number of matches 26893
Number of matches After Lowe's Ratio 7274
Number of Robust matches 3998

 34%|[redacted]        | 21/61 [01:33<03:02,  4.56s/it]
Number of matches 26851
Number of matches After Lowe's Ratio 6525
Number of Robust matches 3239

 36%|[redacted]        | 22/61 [01:37<02:53,  4.46s/it]
Number of matches 25810
Number of matches After Lowe's Ratio 7079
Number of Robust matches 3751

 38%|[redacted]        | 23/61 [01:41<02:48,  4.45s/it]
Number of matches 26666
Number of matches After Lowe's Ratio 6600
Number of Robust matches 3170

 39%|[redacted]        | 24/61 [01:46<02:43,  4.42s/it]
Number of matches 26905
Number of matches After Lowe's Ratio 6179
Number of Robust matches 2722

 41%|[redacted]        | 25/61 [01:50<02:37,  4.38s/it]
Number of matches 29565
Number of matches After Lowe's Ratio 4133
Number of Robust matches 1444

 43%|[redacted]        | 26/61 [01:55<02:42,  4.66s/it]
Number of matches 26852
Number of matches After Lowe's Ratio 4778
Number of Robust matches 1505

 44%|[redacted]        | 27/61 [02:00<02:34,  4.55s/it]
```

```
Number of matches 26187
Number of matches After Lowe's Ratio 5379
Number of Robust matches 1630

 46%|[redacted]      | 28/61 [02:04<02:29,  4.54s/it]

Number of matches 28473
Number of matches After Lowe's Ratio 5415
Number of Robust matches 1358

 48%|[redacted]      | 29/61 [02:09<02:26,  4.59s/it]

Number of matches 26180
Number of matches After Lowe's Ratio 3120
Number of Robust matches 1118

 49%|[redacted]      | 30/61 [02:13<02:19,  4.52s/it]

Number of matches 27047
Number of matches After Lowe's Ratio 4908
Number of Robust matches 1859

 51%|[redacted]      | 31/61 [02:18<02:14,  4.50s/it]

Number of matches 25270
Number of matches After Lowe's Ratio 2591
Number of Robust matches 1010

 52%|[redacted]      | 32/61 [02:21<02:04,  4.30s/it]

Number of matches 24696
Number of matches After Lowe's Ratio 958
Number of Robust matches 243

 54%|[redacted]      | 33/61 [02:25<01:57,  4.20s/it]

Number of matches 25191
Number of matches After Lowe's Ratio 4979
Number of Robust matches 2258

 56%|[redacted]      | 34/61 [02:30<01:54,  4.25s/it]

Number of matches 23938
Number of matches After Lowe's Ratio 5885
Number of Robust matches 2903

 57%|[redacted]      | 35/61 [02:33<01:45,  4.07s/it]

Number of matches 25262
Number of matches After Lowe's Ratio 5287
Number of Robust matches 2650

 59%|[redacted]      | 36/61 [02:38<01:43,  4.13s/it]

Number of matches 26751
Number of matches After Lowe's Ratio 6252
Number of Robust matches 2607
```

```
 61%|      | 37/61 [02:42<01:41,  4.23s/it]

Number of matches 28894
Number of matches After Lowe's Ratio 4637
Number of Robust matches 2004


 62%|      | 38/61 [02:47<01:42,  4.47s/it]

Number of matches 29957
Number of matches After Lowe's Ratio 5407
Number of Robust matches 1796


 64%|      | 39/61 [02:52<01:40,  4.58s/it]

Number of matches 28961
Number of matches After Lowe's Ratio 4810
Number of Robust matches 1870


 66%|      | 40/61 [02:57<01:37,  4.62s/it]

Number of matches 27849
Number of matches After Lowe's Ratio 6265
Number of Robust matches 2679


 67%|      | 41/61 [03:02<01:35,  4.78s/it]

Number of matches 27474
Number of matches After Lowe's Ratio 6678
Number of Robust matches 3207


 69%|      | 42/61 [03:06<01:27,  4.62s/it]

Number of matches 26655
Number of matches After Lowe's Ratio 6971
Number of Robust matches 3735


 70%|      | 43/61 [03:11<01:22,  4.59s/it]

Number of matches 27144
Number of matches After Lowe's Ratio 8105
Number of Robust matches 4169


 72%|      | 44/61 [03:15<01:17,  4.57s/it]

Number of matches 28394
Number of matches After Lowe's Ratio 7407
Number of Robust matches 3912


 74%|      | 45/61 [03:20<01:14,  4.68s/it]

Number of matches 29237
Number of matches After Lowe's Ratio 7815
Number of Robust matches 4253


 75%|      | 46/61 [03:25<01:11,  4.75s/it]
```

```
Number of matches 28652
Number of matches After Lowe's Ratio 8135
Number of Robust matches 4072
```

```
 77%|[redacted]  | 47/61 [03:30<01:05,  4.70s/it]
```

```
Number of matches 28667
Number of matches After Lowe's Ratio 7430
Number of Robust matches 3680
```

```
 79%|[redacted]  | 48/61 [03:35<01:03,  4.89s/it]
```

```
Number of matches 27485
Number of matches After Lowe's Ratio 4449
Number of Robust matches 2339
```

```
 80%|[redacted]  | 49/61 [03:39<00:57,  4.77s/it]
```

```
Number of matches 26460
Number of matches After Lowe's Ratio 8202
Number of Robust matches 4834
```

```
 82%|[redacted]  | 50/61 [03:44<00:51,  4.67s/it]
```

```
Number of matches 27417
Number of matches After Lowe's Ratio 8692
Number of Robust matches 5233
```

```
 84%|[redacted]  | 51/61 [03:48<00:45,  4.54s/it]
```

```
Number of matches 25103
Number of matches After Lowe's Ratio 5108
Number of Robust matches 2843
```

```
 85%|[redacted]  | 52/61 [03:52<00:39,  4.38s/it]
```

```
Number of matches 25624
Number of matches After Lowe's Ratio 5325
Number of Robust matches 2960
```

```
 87%|[redacted]  | 53/61 [03:56<00:34,  4.34s/it]
```

```
Number of matches 26328
Number of matches After Lowe's Ratio 7116
Number of Robust matches 3504
```

```
 89%|[redacted]  | 54/61 [04:00<00:29,  4.24s/it]
```

```
Number of matches 25673
Number of matches After Lowe's Ratio 5013
Number of Robust matches 2532
```

```
 90%|[redacted]  | 55/61 [04:05<00:25,  4.30s/it]
```

```
Number of matches 24822
Number of matches After Lowe's Ratio 5337
Number of Robust matches 2309
```

```
 92%|██████████| 56/61 [04:09<00:21,  4.22s/it]
```

```
Number of matches 24959
Number of matches After Lowe's Ratio 4916
Number of Robust matches 2205
```

```
 93%|██████████| 57/61 [04:13<00:16,  4.14s/it]
```

```
Number of matches 26479
Number of matches After Lowe's Ratio 7173
Number of Robust matches 2994
```

```
 95%|██████████| 58/61 [04:17<00:12,  4.26s/it]
```

```
Number of matches 26722
Number of matches After Lowe's Ratio 4063
Number of Robust matches 1121
```

```
 97%|██████████| 59/61 [04:22<00:08,  4.30s/it]
```

```
Number of matches 26875
Number of matches After Lowe's Ratio 6246
Number of Robust matches 1783
```

```
 98%|██████████| 60/61 [04:26<00:04,  4.45s/it]
  0%|          | 0/40 [00:00<?, ?it/s]
```

```
Number of matches 25965
Number of matches After Lowe's Ratio 1585
Number of Robust matches 439
```

```
  2%|          | 1/40 [00:04<02:51,  4.39s/it]
```

```
Number of matches 26529
Number of matches After Lowe's Ratio 3298
Number of Robust matches 1556
```

```
  5%|          | 2/40 [00:08<02:43,  4.30s/it]
```

```
Number of matches 28060
Number of matches After Lowe's Ratio 5457
Number of Robust matches 3210
```

```
  8%|          | 3/40 [00:13<02:52,  4.66s/it]
```

```
Number of matches 24832
Number of matches After Lowe's Ratio 5300
Number of Robust matches 2661
```

```
 10%|          | 4/40 [00:17<02:34,  4.29s/it]
```

```
Number of matches 20333
Number of matches After Lowe's Ratio 2605
Number of Robust matches 1244
```

```
 12%|▉         | 5/40 [00:20<02:16,  3.91s/it]
```

```
Number of matches 23786
Number of matches After Lowe's Ratio 1621
Number of Robust matches 799
```

```
 15%|▉         | 6/40 [00:24<02:11,  3.87s/it]
```

```
Number of matches 21551
Number of matches After Lowe's Ratio 5447
Number of Robust matches 2926
```

```
 18%|▉         | 7/40 [00:28<02:04,  3.76s/it]
```

```
Number of matches 29428
Number of matches After Lowe's Ratio 3261
Number of Robust matches 1641
```

```
 20%|▉         | 8/40 [00:33<02:14,  4.20s/it]
```

```
Number of matches 27940
Number of matches After Lowe's Ratio 9838
Number of Robust matches 6105
```

```
 22%|▉         | 9/40 [00:37<02:14,  4.33s/it]
```

```
Number of matches 29235
Number of matches After Lowe's Ratio 9987
Number of Robust matches 6396
```

```
 25%|▉         | 10/40 [00:42<02:13,  4.44s/it]
```

```
Number of matches 25475
Number of matches After Lowe's Ratio 7332
Number of Robust matches 4783
```

```
 28%|▉         | 11/40 [00:47<02:14,  4.63s/it]
```

```
Number of matches 25710
Number of matches After Lowe's Ratio 8177
Number of Robust matches 4729
```

```
 30%|▉         | 12/40 [00:51<02:05,  4.46s/it]
```

```
Number of matches 25303
Number of matches After Lowe's Ratio 5706
Number of Robust matches 2869
```

```
 32%|▉         | 13/40 [00:55<01:59,  4.42s/it]
```

```
Number of matches 26839
Number of matches After Lowe's Ratio 7526
Number of Robust matches 4078
```

```
 35%|▉         | 14/40 [01:00<01:56,  4.49s/it]
```

```
Number of matches 27624
Number of matches After Lowe's Ratio 7280
```

```
Number of matches After Lowe's Ratio 7200
Number of Robust matches 3483

 38%|[redacted]| 15/40 [01:05<01:52,  4.51s/it]

Number of matches 28800
Number of matches After Lowe's Ratio 8038
Number of Robust matches 3300

 40%|[redacted]| 16/40 [01:09<01:50,  4.60s/it]

Number of matches 27816
Number of matches After Lowe's Ratio 7518
Number of Robust matches 3093

 42%|[redacted]| 17/40 [01:14<01:46,  4.61s/it]

Number of matches 28846
Number of matches After Lowe's Ratio 8483
Number of Robust matches 3103

 45%|[redacted]| 18/40 [01:19<01:46,  4.83s/it]

Number of matches 29646
Number of matches After Lowe's Ratio 7556
Number of Robust matches 2564

 48%|[redacted]| 19/40 [01:24<01:41,  4.85s/it]

Number of matches 29031
Number of matches After Lowe's Ratio 9381
Number of Robust matches 3253

 50%|[redacted]| 20/40 [01:29<01:37,  4.87s/it]

Number of matches 25780
Number of matches After Lowe's Ratio 7291
Number of Robust matches 2261

 52%|[redacted]| 21/40 [01:34<01:29,  4.70s/it]

Number of matches 27656
Number of matches After Lowe's Ratio 7211
Number of Robust matches 2458

 55%|[redacted]| 22/40 [01:38<01:24,  4.72s/it]

Number of matches 25457
Number of matches After Lowe's Ratio 5767
Number of Robust matches 2461

 57%|[redacted]| 23/40 [01:43<01:18,  4.60s/it]

Number of matches 29115
Number of matches After Lowe's Ratio 1615
Number of Robust matches 472
```

```
60%|      | 24/40 [01:48<01:16,  4.75s/it]

Number of matches 27792
Number of matches After Lowe's Ratio 3919
Number of Robust matches 1094


62%|      | 25/40 [01:53<01:11,  4.78s/it]

Number of matches 31332
Number of matches After Lowe's Ratio 86
Number of Robust matches 22


65%|      | 26/40 [01:58<01:08,  4.88s/it]

Number of matches 28948
Number of matches After Lowe's Ratio 3323
Number of Robust matches 1023


68%|      | 27/40 [02:03<01:03,  4.87s/it]

Number of matches 27031
Number of matches After Lowe's Ratio 6915
Number of Robust matches 2396


70%|      | 28/40 [02:07<00:56,  4.67s/it]

Number of matches 26151
Number of matches After Lowe's Ratio 5846
Number of Robust matches 2079


72%|      | 29/40 [02:11<00:50,  4.56s/it]

Number of matches 26662
Number of matches After Lowe's Ratio 5383
Number of Robust matches 1457


75%|      | 30/40 [02:16<00:45,  4.56s/it]

Number of matches 30209
Number of matches After Lowe's Ratio 5275
Number of Robust matches 1488


78%|      | 31/40 [02:21<00:43,  4.78s/it]

Number of matches 30044
Number of matches After Lowe's Ratio 5492
Number of Robust matches 1727


80%|      | 32/40 [02:26<00:39,  4.95s/it]

Number of matches 30021
Number of matches After Lowe's Ratio 8996
Number of Robust matches 3116


82%|      | 33/40 [02:31<00:34,  4.92s/it]

Number of matches 27536
```

```
Number of matches After Lowe's Ratio 5105
Number of Robust matches 1895
```

```
 85%|▉▉▉▉▉▉▉▉▉ | 34/40 [02:36<00:29,  4.84s/it]
```

```
Number of matches 25241
Number of matches After Lowe's Ratio 7014
Number of Robust matches 2548
```

```
 88%|▉▉▉▉▉▉▉▉▉▉ | 35/40 [02:40<00:23,  4.67s/it]
```

```
Number of matches 25658
Number of matches After Lowe's Ratio 6186
Number of Robust matches 2412
```

```
 90%|▉▉▉▉▉▉▉▉▉▉ | 36/40 [02:44<00:18,  4.59s/it]
```

```
Number of matches 24494
Number of matches After Lowe's Ratio 4850
Number of Robust matches 2105
```

```
 92%|▉▉▉▉▉▉▉▉▉▉▉ | 37/40 [02:48<00:13,  4.35s/it]
```

```
Number of matches 23320
Number of matches After Lowe's Ratio 3467
Number of Robust matches 2140
```

```
 95%|▉▉▉▉▉▉▉▉▉▉▉▉ | 38/40 [02:52<00:08,  4.18s/it]
```

```
Number of matches 24240
Number of matches After Lowe's Ratio 5181
Number of Robust matches 2866
```

```
 98%|▉▉▉▉▉▉▉▉▉▉▉▉▉| 39/40 [02:56<00:04,  4.53s/it]
```

```
Number of matches 23582
Number of matches After Lowe's Ratio 5028
Number of Robust matches 2640
```

In [ ]:

```
H_left_agast = []
H_right_agast = []

num_matches_agast = []
num_good_matches_agast = []

for j in tqdm(range(len(images_left))):
    if j==len(images_left)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_agast[j:j+2][::-1],points_all_left_agast[j:j+2][::-1],descriptors_all_left_agast[j:j+2]
[::-1])
    H_left_agast.append(H_a)
    num_matches_agast.append(matches)
    num_good_matches_agast.append(gd_matches)

for j in tqdm(range(len(images_right))):
```

```
    if j==len(images_right)-1:
        break

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_agast[j:j+2][::-1],points_all_right_agast[j:j+2][::-1],descriptors_all_right_agast[j:
j+2][::-1])
    H_right_agast.append(H_a)
    num_matches_agast.append(matches)
    num_good_matches_agast.append(gd_matches)
```

In [23]:

```
def warpnImages(images_left, images_right,H_left,H_right):
    #img1-centre,img2-left,img3-right

    h, w = images_left[0].shape[:2]

    pts_left = []
    pts_right = []

    pts_centre = np.float32([[0, 0], [0, h], [w, h], [w, 0]]).reshape(-1, 1, 2)

    for j in range(len(H_left)):
      pts = np.float32([[0, 0], [0, h], [w, h], [w, 0]]).reshape(-1, 1, 2)
      pts_left.append(pts)

    for j in range(len(H_right)):
      pts = np.float32([[0, 0], [0, h], [w, h], [w, 0]]).reshape(-1, 1, 2)
      pts_right.append(pts)

    pts_left_transformed=[]
    pts_right_transformed=[]

    for j,pts in enumerate(pts_left):
      if j==0:
        H_trans = H_left[j]
      else:
        H_trans = H_trans@H_left[j]
      pts_ = cv2.perspectiveTransform(pts, H_trans)
      pts_left_transformed.append(pts_)

    for j,pts in enumerate(pts_right):
      if j==0:
        H_trans = H_right[j]
      else:
        H_trans = H_trans@H_right[j]
      pts_ = cv2.perspectiveTransform(pts, H_trans)
      pts_right_transformed.append(pts_)


    print('Step1:Done')


    #pts = np.concatenate((pts1, pts2_), axis=0)

    pts_concat = np.concatenate((pts_centre,np.concatenate(np.array(pts_left_transformed
),axis=0),np.concatenate(np.array(pts_right_transformed),axis=0)), axis=0)

    [xmin, ymin] = np.int32(pts_concat.min(axis=0).ravel() - 0.5)
    [xmax, ymax] = np.int32(pts_concat.max(axis=0).ravel() + 0.5)
    t = [-xmin, -ymin]
    Ht = np.array([[1, 0, t[0]], [0, 1, t[1]], [0, 0, 1]])  # translate

    print('Step2:Done')


    return xmax,xmin,ymax,ymin,t,h,w,Ht
```

In [24]:

```
def final_steps_left_union(images_left,H_left,xmax,xmin,ymax,ymin,t,h,w,Ht):
```

```
    for j,H in enumerate(H_left):
        if j==  0:
            H_trans = Ht@H

        else:
            H_trans = H_trans@H
        result =  cv2.warpPerspective(images_left[j+1],H_trans,(xmax-xmin,ymax-ymin))
        warp_img_init_curr = result

        if j == 0:
            result[t[1]:h+t[1],t[0]:w+t[0]] = images_left[0]
            warp_img_init_prev = result
            continue
        black_pixels = np.where((warp_img_init_prev[:,:,0]==0)&(warp_img_init_prev[:,:,1
]==0)&(warp_img_init_prev[:,:,2]==0))
        warp_img_init_prev[black_pixels] = warp_img_init_curr[black_pixels]

    print('step31:Done')
    return warp_img_init_prev

def final_step_right_union(warp_img_prev,images_right,H_right,xmax,xmin,ymax,ymin,t,h,w,
Ht):
    for j,H in enumerate(H_right):
        if j==  0:
            H_trans = Ht@H
        else:
            H_trans = H_trans@H
        result =  cv2.warpPerspective(images_right[j+1],H_trans,(xmax-xmin,ymax-ymin))
        warp_img_init_curr = result



        black_pixels = np.where((warp_img_prev[:,:,0]==0)&(warp_img_prev[:,:,1]==0)&(war
p_img_prev[:,:,2]==0))
        warp_img_prev[black_pixels] = warp_img_init_curr[black_pixels]

    print('step32:Done')
    return warp_img_prev
```

In [23]:

```
xmax,xmin,ymax,ymin,t,h,w,Ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_brisk,H_right_brisk)
```

```
Step1:Done
Step2:Done
```

In [24]:

```
warp_imgs_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_brisk,xmax,xmin
,ymax,ymin,t,h,w,Ht)
```

```
step31:Done
```

In [25]:

```
warp_imgs_all_brisk = final_step_right_union(warp_imgs_left,images_right_bgr_no_enhance,H
_right_brisk,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

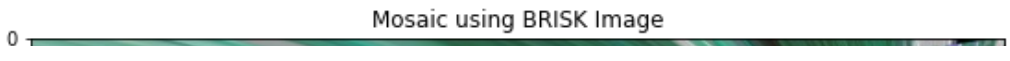
```
step32:Done
```

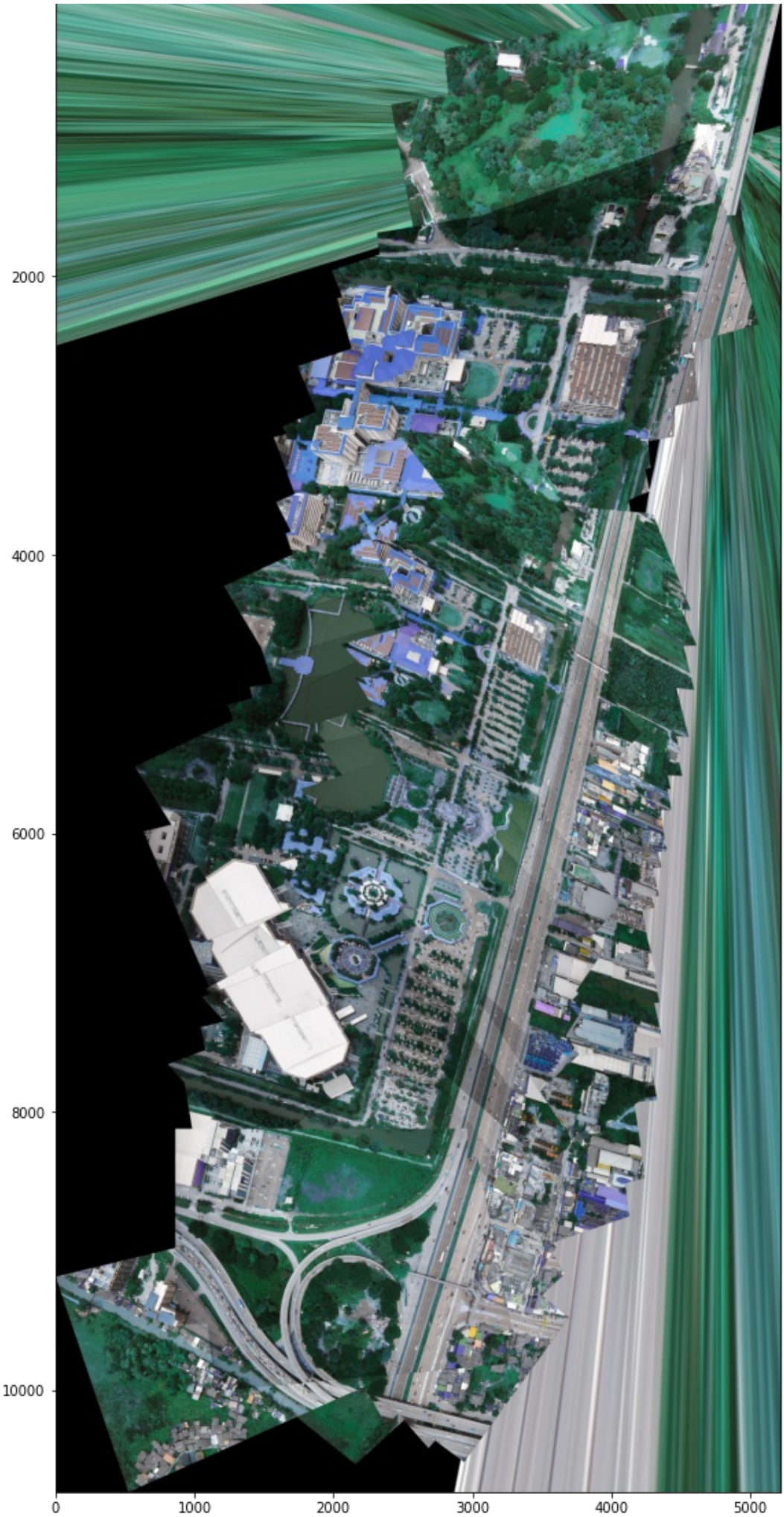
In [26]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_brisk)
plt.title('Mosaic using BRISK Image')
```

Out[26]:

```
Text(0.5, 1.0, 'Mosaic using BRISK Image')
```





In [27]:

```
xmax,xmin,ymax,ymin,t,h,w,Ht =warpnImages(images_left_bgr_no_enhance, images_right_bgr_n
o_enhance,H_left_surfsift,H_right_surfsift)
```

```
Step1:Done
Step2:Done
```

In [42]:

```
warp_imgs_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_surfsift,xmax,x
min,ymax,ymin,t,h,w,Ht)
```

```
step31:Done
```

In [43]:

```
warp_imgs_all_surfsift = final_step_right_union(warp_imgs_left,images_right_bgr_no_enhanc
e,H_right_surfsift,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

```
step32:Done
```

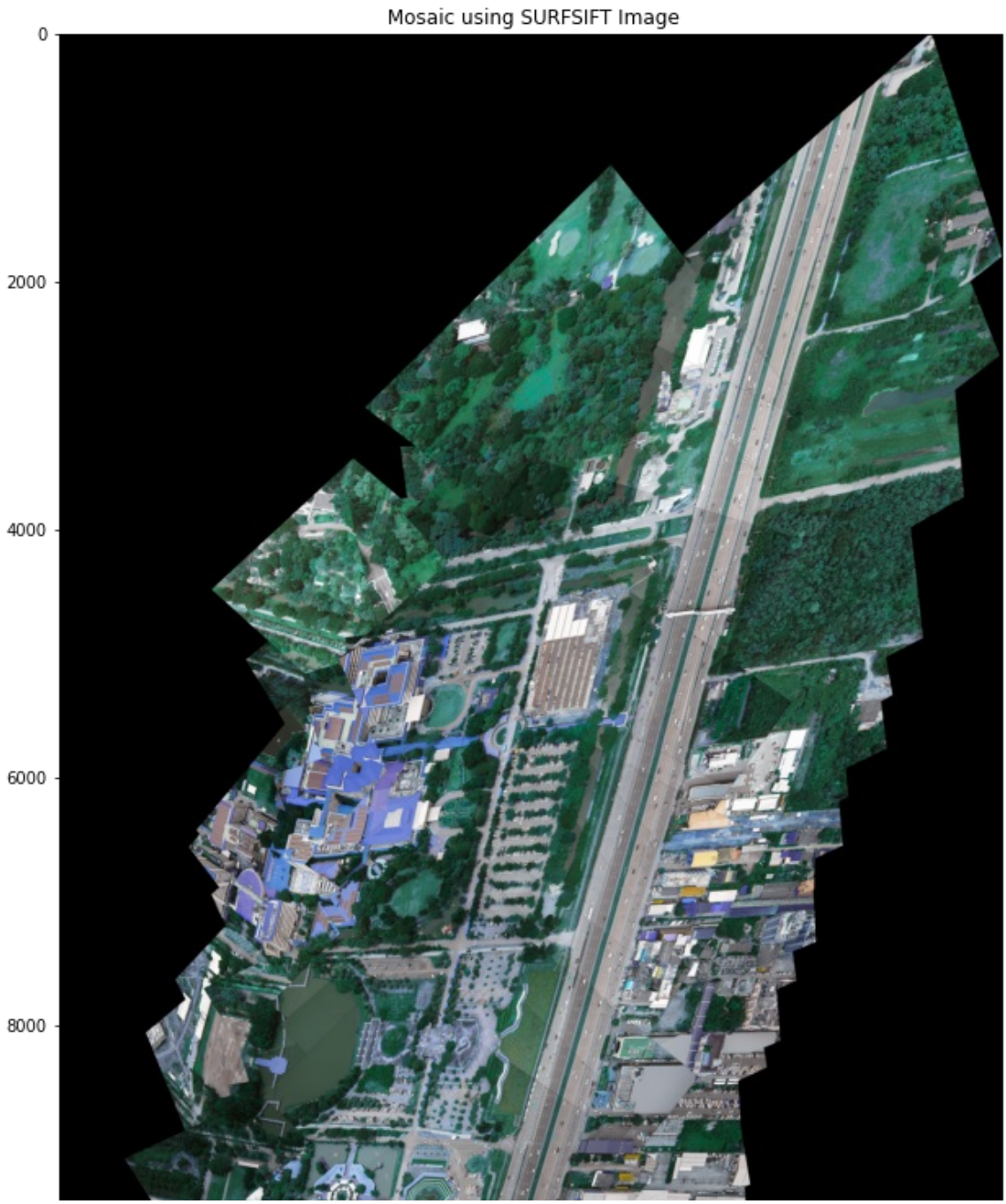
In [44]:

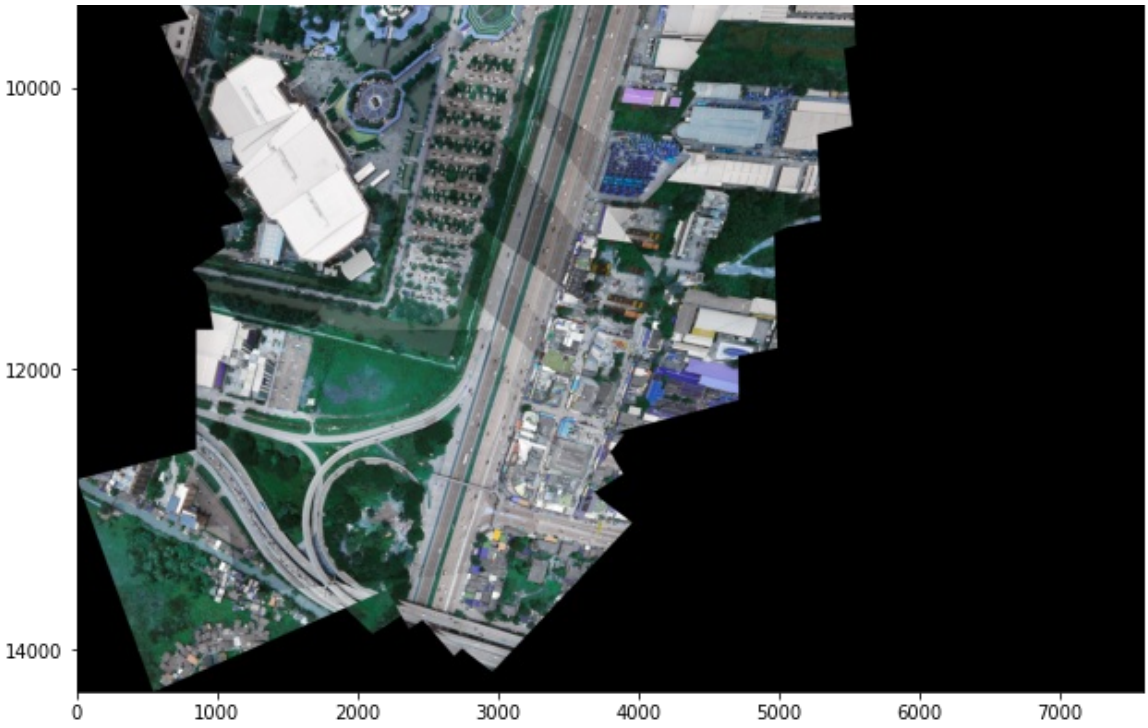
```
plt.figure(figsize=(20,20))

plt.imshow(warp_imgs_all_surfsift)
plt.title(' Mosaic using SURFSIFT Image')
```

Out[44]:

```
Text(0.5, 1.0, ' Mosaic using SURFSIFT Image')
```





In [23]:

```
omax,omin,umax,umin,T,H,W,HT = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_orb,H_right_orb)
```

```
Step1:Done
Step2:Done
```

In [24]:

```
warp_img = final_steps_left_union(images_left_bgr_no_enhance,H_left_orb,omax,omin,umax,um
in,T,H,W,HT)
```

```
step31:Done
```

In [25]:

```
warp_imgs_all_orb = final_step_right_union(warp_img,images_right_bgr_no_enhance,H_right_o
rb,omax,omin,umax,umin,T,H,W,HT)
```

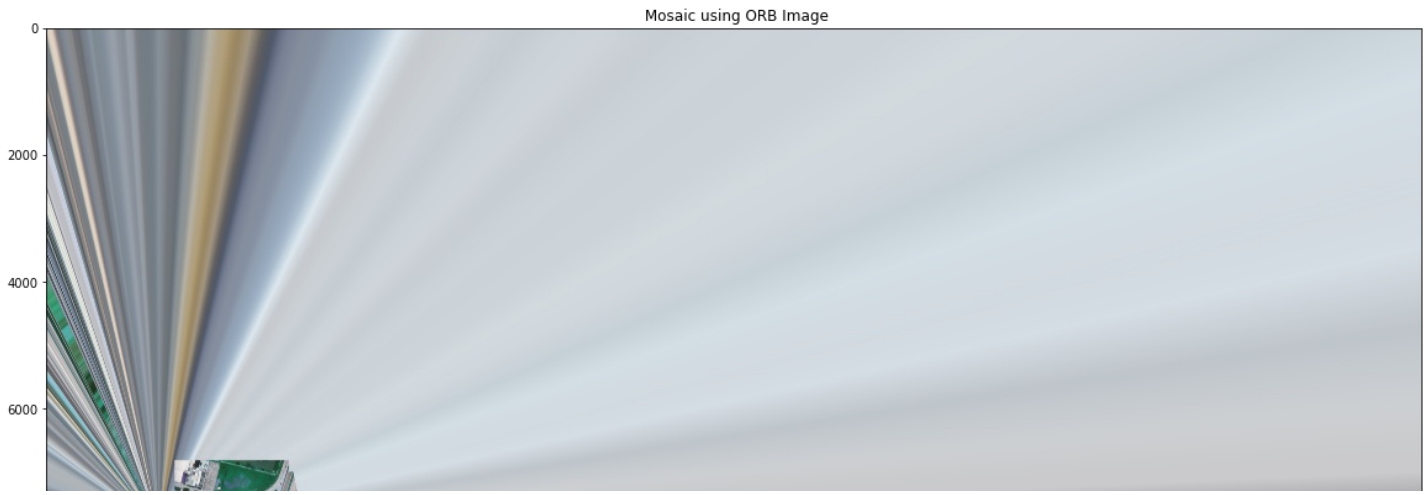
```
step32:Done
```

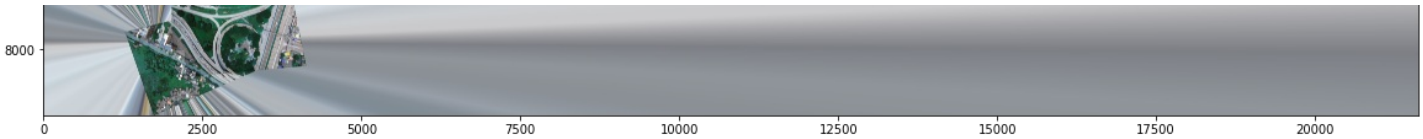
In [26]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_orb)
plt.title(' Mosaic using ORB Image')
```

Out[26]:

```
Text(0.5, 1.0, ' Mosaic using ORB Image')
```





In [25]:

```
mmax,mmin,nmax,nmin,d,e,f,g = warpnImages(images_left_bgr_no_enhance, images_right_bgr_n
o_enhance,H_left_fast,H_right_fast)
```

```
Step1:Done
Step2:Done
```

In [ ]:

```
warp_imgs_fast = final_steps_left_union(images_left_bgr_no_enhance,H_left_fast,mmax,mmin,
nmax,nmin,d,e,f,g)
```

In [ ]:

```
warp_imgs_all_fast = final_step_right_union(warp_imgs_fast,images_right_bgr_no_enhance,H_
right_fast,mmax,mmin,nmax,nmin,d,e,f,g)
```

In [ ]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_fast)
plt.title(' Mosaic using FAST Image')
```

In [ ]:

```
omax,omin,umax,umin,T,H,W,HT = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_akaze,H_right_akaze)
```

In [ ]:

```
warp_img_kaze = final_steps_left_union(images_left_bgr_no_enhance,H_left_akaze,omax,omin,
umax,umin,T,H,W,HT)
```

In [ ]:

```
warp_imgs_all_akaze = final_step_right_union(warp_img_kaze,images_right_bgr_no_enhance,H_
right_akaze,omax,omin,umax,umin,T,H,W,HT)
```

In [ ]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_akaze)
plt.title('Mosaic using Akaze Image')
```

In [ ]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_freak,H_right_freak)
```

In [ ]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_freak,amax,ami
n,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
warp_imgs_all_gftt = final_step_right_union(warp_image_left,images_right_bgr_no_enhance,H
_right_freak,amax,amin,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_gftt)
```

```
plt.title('Mosaic using FREAK image')
```

In [ ]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_fast,H_right_fast)
```

In [ ]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_fast,amax,amin
,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
warp_imgs_all_agast = final_step_right_union(warp_image_left,images_right_bgr_no_enhance,
H_right_fast,amax,amin,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_fast)
plt.title('Mosaic using FAST image')
```

In [ ]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_agast,H_right_agast)
```

In [ ]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_agast,amax,ami
n,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
warp_imgs_all_agast = final_step_right_union(warp_image_left,images_right_bgr_no_enhance,
H_right_agast,amax,amin,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_agast)
plt.title('Mosaic using AGAST image')
```

In [ ]:

```

```