

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)
    imglindex = np.empty((keypointlength), dtype=np.int16)
    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
            imglindex[index] = j
            index = index + 1
    return [cv2.DMatch(imglindex[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)
    |████████████████████| 25.0 MB 21.7 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-packages (from 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)
    |████████████████████| 30.6 MB 22.8 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-packages (from 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 [7]:

```
gridsize = 8
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.INTER_CUBIC)
    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.INTER_CUBIC)
images_right.append(cv2.cvtColor(right_img, cv2.COLOR_BGR2GRAY).astype('float32')/255
.)
images_right_bgr.append(right_img)

```

```

100%|██████████| 61/61 [01:05<00:00, 1.08s/it]
100%|██████████| 40/40 [00:43<00:00, 1.08s/it]

```

In [8]:

```

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=0.35, fy=0.35, interpolation = cv2.INTER_CUBIC)
    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=0.35, fy=0.35, interpolation = cv2.INTER_CUBIC)
    images_right_bgr_no_enhance.append(right_img)

```

```

100%|██████████| 61/61 [00:24<00:00, 2.51it/s]
100%|██████████| 40/40 [00:16<00:00, 2.48it/s]

```

In [9]:

```

Thresh1=60;
Octaves=8;
#PatternScales=1.0f;
brisk = cv2.BRISK_create(Thresh1,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 [00:55<00:00, 1.09it/s]
100%|██████████| 40/40 [00:35<00:00, 1.13it/s]

```

In []:

```

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):
    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):
    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]))

```

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 [9]:

```
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 []:

```
Thresh1=60;
Octaves=8;
#PatternScales=1.0f;
brisk = cv2.BRISK_create(Thresh1, 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 [07:26<00:00, 7.32s/it]
100%|██████████| 40/40 [05:07<00:00, 7.68s/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 [10]:

```

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 [16:40<00:00, 16.40s/it]
100%|██████████| 40/40 [11:22<00:00, 17.07s/it]

```

In [10]:

```

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]))

```

```

100%|██████████| 61/61 [02:20<00:00, 2.31s/it]
100%|██████████| 40/40 [01:36<00:00, 2.41s/it]

```

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'
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[1,:] >= (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, 202691.25it/s]

In []:

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

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, 391520.06it/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 [16]:

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

```
100%|██████████| 101/101 [00:00<00:00, 219836.38it/s]
```

In []:

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

In [16]:

```
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, 218588.60it/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
    reshould = 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_lfl_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_lfl_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<01:45, 1.75s/it]

Number of matches 25059
Number of matches After Lowe's Ratio 851
Number of Robust matches 317

3%|█| 2/61 [00:03<01:52, 1.91s/it]

Number of matches 30921
Number of matches After Lowe's Ratio 753
Number of Robust matches 216

5%|█| 3/61 [00:06<02:04, 2.15s/it]

Number of matches 26028
Number of matches After Lowe's Ratio 382
Number of Robust matches 6

7%|█| 4/61 [00:08<02:07, 2.24s/it]

Number of matches 23435
Number of matches After Lowe's Ratio 1554
Number of Robust matches 855

8%|█| 5/61 [00:10<01:57, 2.09s/it]

Number of matches 28302
Number of matches After Lowe's Ratio 1953
Number of Robust matches 1036

10%|█| 6/61 [00:12<01:57, 2.14s/it]

Number of matches 26534
Number of matches After Lowe's Ratio 1672
Number of Robust matches 758

11%|█| 7/61 [00:15<02:04, 2.30s/it]

Number of matches 32280
Number of matches After Lowe's Ratio 2201
Number of Robust matches 1198

13%|█| 8/61 [00:17<02:08, 2.43s/it]

Number of matches 22854
Number of matches After Lowe's Ratio 1065
Number of Robust matches 485

15%|██████ | 9/61 [00:20<02:01, 2.33s/it]

Number of matches 31108
Number of matches After Lowe's Ratio 1732
Number of Robust matches 1113

16%|██████ | 10/61 [00:22<02:01, 2.38s/it]

Number of matches 26430
Number of matches After Lowe's Ratio 1225
Number of Robust matches 678

18%|██████ | 11/61 [00:24<01:57, 2.34s/it]

Number of matches 32653
Number of matches After Lowe's Ratio 2148
Number of Robust matches 1390

20%|██████ | 12/61 [00:27<02:00, 2.46s/it]

Number of matches 32146
Number of matches After Lowe's Ratio 2378
Number of Robust matches 1690

21%|██████ | 13/61 [00:30<02:06, 2.64s/it]

Number of matches 37734
Number of matches After Lowe's Ratio 2442
Number of Robust matches 1626

23%|██████ | 14/61 [00:34<02:16, 2.91s/it]

Number of matches 37547
Number of matches After Lowe's Ratio 3446
Number of Robust matches 2588

25%|██████ | 15/61 [00:37<02:18, 3.01s/it]

Number of matches 34839
Number of matches After Lowe's Ratio 2724
Number of Robust matches 1859

26%|██████ | 16/61 [00:40<02:17, 3.06s/it]

Number of matches 30911
Number of matches After Lowe's Ratio 2728
Number of Robust matches 2097

28%|██████ | 17/61 [00:43<02:09, 2.94s/it]

Number of matches 32047
Number of matches After Lowe's Ratio 2532
Number of Robust matches 1830

30%|██████ | 18/61 [00:46<02:04, 2.89s/it]

Number of matches 33078

Number of matches After Lowe's Ratio 3006
Number of Robust matches 2200

31% | ████████ | 19/61 [00:49<02:05, 2.98s/it]

Number of matches 31280
Number of matches After Lowe's Ratio 3171
Number of Robust matches 2300

33% | ████████ | 20/61 [00:51<01:59, 2.90s/it]

Number of matches 31427
Number of matches After Lowe's Ratio 2403
Number of Robust matches 1709

34% | ████████ | 21/61 [00:54<01:52, 2.80s/it]

Number of matches 32135
Number of matches After Lowe's Ratio 1914
Number of Robust matches 1170

36% | ████████ | 22/61 [00:57<01:49, 2.80s/it]

Number of matches 31815
Number of matches After Lowe's Ratio 2139
Number of Robust matches 1353

38% | ████████ | 23/61 [00:59<01:44, 2.75s/it]

Number of matches 33078
Number of matches After Lowe's Ratio 2292
Number of Robust matches 1465

39% | ████████ | 24/61 [01:02<01:44, 2.84s/it]

Number of matches 35870
Number of matches After Lowe's Ratio 1929
Number of Robust matches 1288

41% | ████████ | 25/61 [01:06<01:48, 3.01s/it]

Number of matches 43958
Number of matches After Lowe's Ratio 2438
Number of Robust matches 1029

43% | ████████ | 26/61 [01:10<01:55, 3.29s/it]

Number of matches 37987
Number of matches After Lowe's Ratio 1902
Number of Robust matches 903

44% | ████████ | 27/61 [01:13<01:53, 3.34s/it]

Number of matches 32058
Number of matches After Lowe's Ratio 1801
Number of Robust matches 958

46%|██████ | 28/61 [01:16<01:43, 3.13s/it]

Number of matches 30297
Number of matches After Lowe's Ratio 1544
Number of Robust matches 595

48%|██████ | 29/61 [01:19<01:38, 3.07s/it]

Number of matches 33725
Number of matches After Lowe's Ratio 1033
Number of Robust matches 421

49%|██████ | 30/61 [01:22<01:34, 3.05s/it]

Number of matches 34224
Number of matches After Lowe's Ratio 1381
Number of Robust matches 583

51%|██████ | 31/61 [01:25<01:32, 3.10s/it]

Number of matches 34530
Number of matches After Lowe's Ratio 670
Number of Robust matches 220

52%|██████ | 32/61 [01:28<01:26, 2.99s/it]

Number of matches 24599
Number of matches After Lowe's Ratio 393
Number of Robust matches 40

54%|██████ | 33/61 [01:30<01:15, 2.71s/it]

Number of matches 24704
Number of matches After Lowe's Ratio 1383
Number of Robust matches 620

56%|██████ | 34/61 [01:32<01:05, 2.44s/it]

Number of matches 21196
Number of matches After Lowe's Ratio 1368
Number of Robust matches 694

57%|██████ | 35/61 [01:33<00:57, 2.19s/it]

Number of matches 26252
Number of matches After Lowe's Ratio 1349
Number of Robust matches 672

59%|██████ | 36/61 [01:36<00:55, 2.24s/it]

Number of matches 32005
Number of matches After Lowe's Ratio 1752
Number of Robust matches 785

61%|██████ | 37/61 [01:39<00:59, 2.50s/it]

Number of matches 44696

Number of matches After Lowe's Ratio 1777
Number of Robust matches 584

62%|██████ | 38/61 [01:43<01:11, 3.10s/it]

Number of matches 48468
Number of matches After Lowe's Ratio 2363
Number of Robust matches 681

64%|██████ | 39/61 [01:48<01:21, 3.71s/it]

Number of matches 44596
Number of matches After Lowe's Ratio 2196
Number of Robust matches 836

66%|██████ | 40/61 [01:53<01:22, 3.95s/it]

Number of matches 35503
Number of matches After Lowe's Ratio 2142
Number of Robust matches 1055

67%|██████ | 41/61 [01:56<01:14, 3.71s/it]

Number of matches 32924
Number of matches After Lowe's Ratio 2510
Number of Robust matches 1427

69%|██████ | 42/61 [01:59<01:06, 3.49s/it]

Number of matches 30917
Number of matches After Lowe's Ratio 2593
Number of Robust matches 1748

70%|██████ | 43/61 [02:02<00:57, 3.20s/it]

Number of matches 30511
Number of matches After Lowe's Ratio 2736
Number of Robust matches 1763

72%|██████ | 44/61 [02:04<00:52, 3.06s/it]

Number of matches 36828
Number of matches After Lowe's Ratio 2546
Number of Robust matches 1505

74%|██████ | 45/61 [02:08<00:50, 3.14s/it]

Number of matches 39748
Number of matches After Lowe's Ratio 3058
Number of Robust matches 1635

75%|██████ | 46/61 [02:12<00:51, 3.41s/it]

Number of matches 38033
Number of matches After Lowe's Ratio 3039
Number of Robust matches 1538

77%|██████████ | 47/61 [02:15<00:48, 3.44s/it]

Number of matches 39826
Number of matches After Lowe's Ratio 3038
Number of Robust matches 1787

79%|██████████ | 48/61 [02:19<00:44, 3.44s/it]

Number of matches 33296
Number of matches After Lowe's Ratio 1982
Number of Robust matches 1140

80%|██████████ | 49/61 [02:22<00:40, 3.37s/it]

Number of matches 31758
Number of matches After Lowe's Ratio 3476
Number of Robust matches 2427

82%|██████████ | 50/61 [02:25<00:35, 3.22s/it]

Number of matches 31028
Number of matches After Lowe's Ratio 3040
Number of Robust matches 2136

84%|██████████ | 51/61 [02:27<00:30, 3.01s/it]

Number of matches 27881
Number of matches After Lowe's Ratio 1741
Number of Robust matches 918

85%|██████████ | 52/61 [02:29<00:25, 2.80s/it]

Number of matches 27362
Number of matches After Lowe's Ratio 1739
Number of Robust matches 1003

87%|██████████ | 53/61 [02:32<00:21, 2.67s/it]

Number of matches 27680
Number of matches After Lowe's Ratio 2302
Number of Robust matches 1383

89%|██████████ | 54/61 [02:34<00:17, 2.54s/it]

Number of matches 32777
Number of matches After Lowe's Ratio 2219
Number of Robust matches 1141

90%|██████████ | 55/61 [02:37<00:15, 2.60s/it]

Number of matches 27042
Number of matches After Lowe's Ratio 2067
Number of Robust matches 1428

92%|██████████ | 56/61 [02:39<00:12, 2.48s/it]

Number of matches 20160

Number of matches 29160
Number of matches After Lowe's Ratio 1868
Number of Robust matches 859

93%|██████████| 57/61 [02:42<00:10, 2.53s/it]

Number of matches 32830
Number of matches After Lowe's Ratio 2815
Number of Robust matches 1293

95%|██████████| 58/61 [02:45<00:07, 2.64s/it]

Number of matches 33004
Number of matches After Lowe's Ratio 1828
Number of Robust matches 707

97%|██████████| 59/61 [02:47<00:05, 2.71s/it]

Number of matches 35929
Number of matches After Lowe's Ratio 2770
Number of Robust matches 1062

98%|██████████| 60/61 [02:50<00:02, 2.85s/it]
0%|██████████| 0/40 [00:00<?, ?it/s]

Number of matches 25316
Number of matches After Lowe's Ratio 808
Number of Robust matches 219

2%|██████████| 1/40 [00:01<01:17, 2.00s/it]

Number of matches 24778
Number of matches After Lowe's Ratio 918
Number of Robust matches 445

5%|██████████| 2/40 [00:04<01:16, 2.02s/it]

Number of matches 35128
Number of matches After Lowe's Ratio 1894
Number of Robust matches 1294

8%|██████████| 3/40 [00:07<01:31, 2.48s/it]

Number of matches 28715
Number of matches After Lowe's Ratio 2188
Number of Robust matches 1449

10%|██████████| 4/40 [00:09<01:31, 2.54s/it]

Number of matches 25864
Number of matches After Lowe's Ratio 1212
Number of Robust matches 752

12%|██████████| 5/40 [00:11<01:21, 2.33s/it]

Number of matches 24371
Number of matches After Lowe's Ratio 626
Number of Robust matches 253

15%|██████ | 6/40 [00:13<01:15, 2.22s/it]

Number of matches 21250
Number of matches After Lowe's Ratio 1628
Number of Robust matches 1161

18%|██████ | 7/40 [00:15<01:06, 2.02s/it]

Number of matches 31082
Number of matches After Lowe's Ratio 1070
Number of Robust matches 567

20%|██████ | 8/40 [00:17<01:11, 2.22s/it]

Number of matches 31880
Number of matches After Lowe's Ratio 2705
Number of Robust matches 2118

22%|██████ | 9/40 [00:20<01:12, 2.35s/it]

Number of matches 33259
Number of matches After Lowe's Ratio 2998
Number of Robust matches 2294

25%|██████ | 10/40 [00:23<01:13, 2.47s/it]

Number of matches 29687
Number of matches After Lowe's Ratio 2439
Number of Robust matches 1932

28%|██████ | 11/40 [00:26<01:14, 2.57s/it]

Number of matches 33001
Number of matches After Lowe's Ratio 2475
Number of Robust matches 1883

30%|██████ | 12/40 [00:28<01:13, 2.64s/it]

Number of matches 33071
Number of matches After Lowe's Ratio 1844
Number of Robust matches 1251

32%|██████ | 13/40 [00:31<01:13, 2.73s/it]

Number of matches 35123
Number of matches After Lowe's Ratio 2205
Number of Robust matches 1555

35%|██████ | 14/40 [00:35<01:15, 2.91s/it]

Number of matches 39155
Number of matches After Lowe's Ratio 2204
Number of Robust matches 1483

38%|██████ | 15/40 [00:38<01:17, 3.12s/it]

Number of matches 37668
Number of matches After Lowe's Ratio 2543
Number of Robust matches 1490

40% | ████████ | 16/40 [00:42<01:21, 3.41s/it]

Number of matches 40615
Number of matches After Lowe's Ratio 2822
Number of Robust matches 1181

42% | ████████ | 17/40 [00:46<01:21, 3.53s/it]

Number of matches 35723
Number of matches After Lowe's Ratio 2626
Number of Robust matches 1269

45% | ████████ | 18/40 [00:49<01:15, 3.41s/it]

Number of matches 29133
Number of matches After Lowe's Ratio 1990
Number of Robust matches 963

48% | ████████ | 19/40 [00:52<01:04, 3.09s/it]

Number of matches 31575
Number of matches After Lowe's Ratio 2555
Number of Robust matches 901

50% | ████████ | 20/40 [00:54<00:59, 2.96s/it]

Number of matches 30053
Number of matches After Lowe's Ratio 2148
Number of Robust matches 665

52% | ████████ | 21/40 [00:57<00:54, 2.88s/it]

Number of matches 24008
Number of matches After Lowe's Ratio 1549
Number of Robust matches 677

55% | ████████ | 22/40 [00:59<00:46, 2.59s/it]

Number of matches 29051
Number of matches After Lowe's Ratio 1407
Number of Robust matches 709

57% | ████████ | 23/40 [01:01<00:43, 2.59s/it]

Number of matches 45494
Number of matches After Lowe's Ratio 604
Number of Robust matches 107

60% | ████████ | 24/40 [01:06<00:50, 3.17s/it]

Number of matches 41453
Number of matches After Lowe's Ratio 1048
Number of Robust matches 408

Number of Robust matches 100

62%|██████████ | 25/40 [01:10<00:53, 3.56s/it]

Number of matches 47391
Number of matches After Lowe's Ratio 362
Number of Robust matches 7

65%|██████████ | 26/40 [01:15<00:55, 3.98s/it]

Number of matches 36711
Number of matches After Lowe's Ratio 1045
Number of Robust matches 353

68%|██████████ | 27/40 [01:19<00:49, 3.81s/it]

Number of matches 34496
Number of matches After Lowe's Ratio 1938
Number of Robust matches 747

70%|██████████ | 28/40 [01:22<00:42, 3.54s/it]

Number of matches 32321
Number of matches After Lowe's Ratio 2031
Number of Robust matches 712

72%|██████████ | 29/40 [01:25<00:36, 3.31s/it]

Number of matches 28452
Number of matches After Lowe's Ratio 1665
Number of Robust matches 582

75%|██████████ | 30/40 [01:27<00:29, 2.97s/it]

Number of matches 26073
Number of matches After Lowe's Ratio 1353
Number of Robust matches 398

78%|██████████ | 31/40 [01:29<00:24, 2.69s/it]

Number of matches 27594
Number of matches After Lowe's Ratio 1314
Number of Robust matches 434

80%|██████████ | 32/40 [01:31<00:21, 2.66s/it]

Number of matches 29922
Number of matches After Lowe's Ratio 2412
Number of Robust matches 742

82%|██████████ | 33/40 [01:34<00:18, 2.60s/it]

Number of matches 31761
Number of matches After Lowe's Ratio 1456
Number of Robust matches 457

85%|██████████ | 34/40 [01:36<00:15, 2.50s/it]

85%|██████████ | 34/40 [01:36<00:15, 2.59s/it]

Number of matches 28598
Number of matches After Lowe's Ratio 2076
Number of Robust matches 1017

88%|██████████ | 35/40 [01:39<00:12, 2.58s/it]

Number of matches 32677
Number of matches After Lowe's Ratio 1955
Number of Robust matches 748

90%|██████████ | 36/40 [01:42<00:10, 2.66s/it]

Number of matches 25600
Number of matches After Lowe's Ratio 1477
Number of Robust matches 621

92%|██████████ | 37/40 [01:44<00:07, 2.45s/it]

Number of matches 28462
Number of matches After Lowe's Ratio 1093
Number of Robust matches 713

95%|██████████ | 38/40 [01:46<00:05, 2.51s/it]

Number of matches 28784
Number of matches After Lowe's Ratio 1415
Number of Robust matches 877

98%|██████████ | 39/40 [01:49<00:02, 2.80s/it]

Number of matches 27866
Number of matches After Lowe's Ratio 1396
Number of Robust matches 957

In []:

```
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_right_orb[j:j+2][::-1],points_all_right_orb[j:j+2][::-1],descriptors_all_right_orb[j:j+2][::-1])
```

```
:-1])
H_right_orb.append(H_a)
num_matches_orb.append(matches)
num_good_matches_orb.append(gd_matches)
```

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_right_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_right_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_right_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_right_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_superpoint[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_right_superpoint[j:j+2][::-1],points_all_right_superpoint[j:j+2][::-1],descriptors_all_right_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_right_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_right_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 [20]:

```
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_right_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 [00:20<20:44, 20.75s/it]

Number of matches 109090
Number of matches After Lowe's Ratio 4687
Number of Robust matches 1951

3%| | 2/61 [00:43<21:17, 21.65s/it]

Number of matches 121549
Number of matches After Lowe's Ratio 1802
Number of Robust matches 658

5%| | 3/61 [01:06<21:42, 22.46s/it]

Number of matches 106987
Number of matches After Lowe's Ratio 135
Number of Robust matches 46

7%| | 4/61 [01:27<20:57, 22.07s/it]


Number of matches 107417
Number of matches After Lowe's Ratio 15298
Number of Robust matches 8741

8%| | 5/61 [01:49<20:34, 22.04s/it]


Number of matches 109972
Number of matches After Lowe's Ratio 1144
Number of Robust matches 441

10%| | 6/61 [02:11<20:12, 22.04s/it]


Number of matches 105684
Number of matches After Lowe's Ratio 4658
Number of Robust matches 2306

11% |  | 7/61 [02:33<19:42, 21.91s/it]


Number of matches 113643
Number of matches After Lowe's Ratio 2288
Number of Robust matches 1270

13% |  | 8/61 [02:55<19:17, 21.85s/it]


Number of matches 85258
Number of matches After Lowe's Ratio 6920
Number of Robust matches 3807

15% |  | 9/61 [03:12<17:45, 20.49s/it]


Number of matches 104129
Number of matches After Lowe's Ratio 6422
Number of Robust matches 3796

16% |  | 10/61 [03:31<17:02, 20.06s/it]


Number of matches 63147
Number of matches After Lowe's Ratio 2614
Number of Robust matches 1614

18% |  | 11/61 [03:45<15:04, 18.09s/it]


Number of matches 87536
Number of matches After Lowe's Ratio 10445
Number of Robust matches 7444

20% |  | 12/61 [04:02<14:31, 17.79s/it]


Number of matches 75857
Number of matches After Lowe's Ratio 582
Number of Robust matches 285

21% |  | 13/61 [04:18<13:44, 17.18s/it]

Number of matches 97150
Number of matches After Lowe's Ratio 3322
Number of Robust matches 2096

23% |  | 14/61 [04:38<14:06, 18.01s/it]

Number of matches 98403
Number of matches After Lowe's Ratio 7763
Number of Robust matches 4492

25% |  | 15/61 [04:58<14:11, 18.52s/it]

Number of matches 94577
Number of matches After Lowe's Ratio 10231
Number of Robust matches 8256

Number of matches 94935
Number of matches After Lowe's Ratio 7225

26%|██████ | 16/61 [05:16<13:53, 18.52s/it]

Number of Robust matches 5366

28%|██████ | 17/61 [05:36<13:54, 18.96s/it]

Number of matches 101108
Number of matches After Lowe's Ratio 12088
Number of Robust matches 9526

Number of matches 103611
Number of matches After Lowe's Ratio 25675
Number of Robust matches 21832

31%|██████ | 19/61 [06:18<14:06, 20.16s/it]

Number of matches 106411
Number of matches After Lowe's Ratio 22975
Number of Robust matches 15016

33%|██████ | 20/61 [06:41<14:09, 20.73s/it]

Number of matches 115231
Number of matches After Lowe's Ratio 13814
Number of Robust matches 9653

34%|██████ | 21/61 [07:04<14:24, 21.60s/it]

Number of matches 117947
Number of matches After Lowe's Ratio 4424
Number of Robust matches 2637

Number of matches 111829
Number of matches After Lowe's Ratio 25003

36%|██████ | 22/61 [07:28<14:28, 22.26s/it]

Number of Robust matches 15683

38%|██████ | 23/61 [07:50<14:03, 22.21s/it]

Number of matches 112111
Number of matches After Lowe's Ratio 4983
Number of Robust matches 2554

39%|██████ | 24/61 [08:13<13:51, 22.48s/it]

Number of matches 116212
Number of matches After Lowe's Ratio 19163
Number of Robust matches 10891

41%|██████ | 25/61 [08:37<13:41, 22.83s/it]

Number of matches 123205
Number of matches After Lowe's Ratio 184
Number of Robust matches 49

43%|██████ | 26/61 [09:00<13:27, 23.08s/it]

Number of matches 109372
Number of matches After Lowe's Ratio 471
Number of Robust matches 126

44%|██████ | 27/61 [09:22<12:51, 22.69s/it]

Number of matches 112914
Number of matches After Lowe's Ratio 12052
Number of Robust matches 6744

46%|██████ | 28/61 [09:45<12:26, 22.64s/it]

Number of matches 109913
Number of matches After Lowe's Ratio 412
Number of Robust matches 112

48%|██████ | 29/61 [10:07<11:59, 22.49s/it]

Number of matches 120423
Number of matches After Lowe's Ratio 217
Number of Robust matches 74

49%|██████ | 30/61 [10:30<11:46, 22.78s/it]

Number of matches 121925
Number of matches After Lowe's Ratio 6831
Number of Robust matches 2900

51%|██████ | 31/61 [10:54<11:30, 23.02s/it]

Number of matches 116705
Number of matches After Lowe's Ratio 4550
Number of Robust matches 1874

52%|██████ | 32/61 [11:17<11:05, 22.96s/it]

Number of matches 109716
Number of matches After Lowe's Ratio 55
Number of Robust matches 8

54%|██████ | 33/61 [11:38<10:30, 22.53s/it]

Number of matches 110779
Number of matches After Lowe's Ratio 20497
Number of Robust matches 11974

56%|██████ | 34/61 [12:01<10:06, 22.45s/it]

Number of matches 109349
Number of matches After Lowe's Ratio 19116
Number of Robust matches 9373

57%|███████ | 35/61 [12:23<09:41, 22.38s/it]

Number of matches 115938
Number of matches After Lowe's Ratio 17269
Number of Robust matches 10666

59%|███████ | 36/61 [12:45<09:19, 22.39s/it]

Number of matches 122855
Number of matches After Lowe's Ratio 14493
Number of Robust matches 8057

Number of matches 133153
Number of matches After Lowe's Ratio 19090

61%|███████ | 37/61 [13:10<09:15, 23.16s/it]

Number of Robust matches 8866

Number of matches 139272
Number of matches After Lowe's Ratio 21607
Number of Robust matches 7878

64%|███████ | 39/61 [14:03<09:08, 24.92s/it]

Number of matches 132310
Number of matches After Lowe's Ratio 20279
Number of Robust matches 9672

66%|███████ | 40/61 [14:28<08:41, 24.83s/it]

Number of matches 117037
Number of matches After Lowe's Ratio 21833
Number of Robust matches 11621

Number of matches 112444
Number of matches After Lowe's Ratio 25674
Number of Robust matches 16728

67%|███████ | 41/61 [14:51<08:08, 24.42s/it]

Number of matches 105817
Number of matches After Lowe's Ratio 25731
Number of Robust matches 18315

69%|███████ | 42/61 [15:13<07:29, 23.67s/it]

Number of matches 103385
Number of matches After Lowe's Ratio 22656

70%|███████ | 43/61 [15:34<06:50, 22.82s/it]

Number of Robust matches 15014

72%|███████ | 44/61 [15:55<06:17, 22.22s/it]

72%|██████████ | 44/61 [15:55<06:17, 22.23s/it]

Number of matches 109002
Number of matches After Lowe's Ratio 21703
Number of Robust matches 14936

74%|██████████ | 45/61 [16:18<05:58, 22.42s/it]

Number of matches 112071
Number of matches After Lowe's Ratio 23175
Number of Robust matches 13655

Number of matches 106802
Number of matches After Lowe's Ratio 24928

75%|██████████ | 46/61 [16:41<05:38, 22.57s/it]

Number of Robust matches 17374

Number of matches 101920
Number of matches After Lowe's Ratio 18195

77%|██████████ | 47/61 [17:02<05:11, 22.26s/it]

Number of Robust matches 12811

79%|██████████ | 48/61 [17:22<04:38, 21.46s/it]

Number of matches 85971
Number of matches After Lowe's Ratio 10175
Number of Robust matches 7275

80%|██████████ | 49/61 [17:39<04:01, 20.09s/it]

Number of matches 81836
Number of matches After Lowe's Ratio 23618
Number of Robust matches 17036

82%|██████████ | 50/61 [17:56<03:31, 19.24s/it]

Number of matches 87852
Number of matches After Lowe's Ratio 18299
Number of Robust matches 13780

Number of matches 90007
Number of matches After Lowe's Ratio 18186

84%|██████████ | 51/61 [18:13<03:06, 18.68s/it]

Number of Robust matches 13715

85%|██████████ | 52/61 [18:31<02:45, 18.34s/it]

Number of matches 89119
Number of matches After Lowe's Ratio 17661
Number of Robust matches 12285

Number of matches 93962

Number of matches After Lowe's Ratio 26032

87%|██████████ | 53/61 [18:49<02:25, 18.16s/it]

Number of Robust matches 17316

89%|██████████ | 54/61 [19:07<02:07, 18.15s/it]

Number of matches 90558
Number of matches After Lowe's Ratio 7701
Number of Robust matches 4727

90%|██████████ | 55/61 [19:25<01:48, 18.10s/it]

Number of matches 89788
Number of matches After Lowe's Ratio 5278
Number of Robust matches 2941

Number of matches 94690
Number of matches After Lowe's Ratio 11590

92%|██████████ | 56/61 [19:43<01:30, 18.14s/it]

Number of Robust matches 6439

Number of matches 98372
Number of matches After Lowe's Ratio 10432

93%|██████████ | 57/61 [20:02<01:13, 18.37s/it]

Number of Robust matches 5256

Number of matches 97482
Number of matches After Lowe's Ratio 5911

95%|██████████ | 58/61 [20:22<00:56, 18.81s/it]

Number of Robust matches 2244

97%|██████████ | 59/61 [20:41<00:38, 19.01s/it]

Number of matches 100849
Number of matches After Lowe's Ratio 12405
Number of Robust matches 5429

98%|██████████ | 60/61 [21:01<00:21, 21.02s/it]
0%| | 0/40 [00:00<?, ?it/s]

Number of matches 92828
Number of matches After Lowe's Ratio 1184
Number of Robust matches 402

2%| | 1/40 [00:21<13:54, 21.41s/it]

Number of matches 108435
Number of matches After Lowe's Ratio 6716
Number of Robust matches 3712

5% | 2/40 [00:43<13:40, 21.59s/it]

Number of matches 123694
Number of matches After Lowe's Ratio 19495
Number of Robust matches 15039

8% | 3/40 [01:06<13:51, 22.48s/it]

Number of matches 96343
Number of matches After Lowe's Ratio 16631
Number of Robust matches 12998

10% | 4/40 [01:23<12:15, 20.44s/it]

Number of matches 54457
Number of matches After Lowe's Ratio 5903
Number of Robust matches 4187

12% | 5/40 [01:35<10:05, 17.29s/it]

Number of matches 74343
Number of matches After Lowe's Ratio 3542
Number of Robust matches 2159

15% | 6/40 [01:49<09:11, 16.22s/it]

Number of matches 57064
Number of matches After Lowe's Ratio 10117
Number of Robust matches 7733

18% | 7/40 [02:04<08:40, 15.77s/it]

Number of matches 104262
Number of matches After Lowe's Ratio 8771
Number of Robust matches 5890

20% | 8/40 [02:25<09:15, 17.35s/it]

Number of matches 105631
Number of matches After Lowe's Ratio 19927
Number of Robust matches 14157

22% | 9/40 [02:47<09:42, 18.80s/it]

Number of matches 108249
Number of matches After Lowe's Ratio 19849
Number of Robust matches 14916

25% | 10/40 [03:09<09:57, 19.91s/it]

Number of matches 106606
Number of matches After Lowe's Ratio 23858
Number of Robust matches 19380

Number of matches 120200
Number of matches After Lowe's Ratio 33314

28%|██████ | 11/40 [03:32<10:00, 20.71s/it]

Number of Robust matches 24288

30%|██████ | 12/40 [03:56<10:10, 21.82s/it]

Number of matches 125528

Number of matches After Lowe's Ratio 9599

Number of Robust matches 5598

Number of matches 129552

Number of matches After Lowe's Ratio 16218

32%|██████ | 13/40 [04:22<10:17, 22.89s/it]

Number of Robust matches 11236

35%|██████ | 14/40 [04:48<10:20, 23.88s/it]

Number of matches 131203

Number of matches After Lowe's Ratio 18912

Number of Robust matches 10988

38%|██████ | 15/40 [05:13<10:10, 24.42s/it]

Number of matches 128349

Number of matches After Lowe's Ratio 14115

Number of Robust matches 7990

Number of matches 125112

Number of matches After Lowe's Ratio 9255

40%|██████ | 16/40 [05:38<09:49, 24.57s/it]

Number of Robust matches 4242

42%|██████ | 17/40 [06:03<09:23, 24.49s/it]

Number of matches 115397

Number of matches After Lowe's Ratio 18604

Number of Robust matches 8952

45%|██████ | 18/40 [06:25<08:46, 23.91s/it]

Number of matches 107757

Number of matches After Lowe's Ratio 14419

Number of Robust matches 6693

48%|██████ | 19/40 [06:46<08:05, 23.11s/it]

Number of matches 108520

Number of matches After Lowe's Ratio 21520

Number of Robust matches 11235

50%|██████ | 20/40 [07:08<07:33, 22.69s/it]

Number of matches 105170

Number of matches After Lowe's Ratio 21858

Number of Robust matches 10581

52%|███████ | 21/40 [07:29<06:59, 22.06s/it]

Number of matches 101842

Number of matches After Lowe's Ratio 6680

Number of Robust matches 3302

55%|███████ | 22/40 [07:49<06:27, 21.53s/it]

Number of matches 114806

Number of matches After Lowe's Ratio 7875

Number of Robust matches 4241

57%|███████ | 23/40 [08:13<06:18, 22.26s/it]

Number of matches 144158

Number of matches After Lowe's Ratio 3097

Number of Robust matches 1190

60%|███████ | 24/40 [08:41<06:22, 23.89s/it]

Number of matches 129327

Number of matches After Lowe's Ratio 13066

Number of Robust matches 5645

62%|███████ | 25/40 [09:07<06:09, 24.62s/it]

Number of matches 150305

Number of matches After Lowe's Ratio 42

Number of Robust matches 9

65%|███████ | 26/40 [09:35<05:57, 25.57s/it]

Number of matches 125780

Number of matches After Lowe's Ratio 10397

Number of Robust matches 4080

68%|███████ | 27/40 [09:59<05:27, 25.23s/it]

Number of matches 122865

Number of matches After Lowe's Ratio 6986

Number of Robust matches 2535

70%|███████ | 28/40 [10:22<04:55, 24.65s/it]

Number of matches 105783

Number of matches After Lowe's Ratio 17620

Number of Robust matches 7370

72%|███████ | 29/40 [10:43<04:18, 23.52s/it]

Number of matches 102138

Number of matches After Lowe's Ratio 5739

Number of Robust matches 2852

75%|██████████ | 30/40 [11:03<03:42, 22.24s/it]

Number of matches 89671
Number of matches After Lowe's Ratio 10343
Number of Robust matches 4174

78%|██████████ | 31/40 [11:21<03:09, 21.10s/it]

Number of matches 96125
Number of matches After Lowe's Ratio 16161
Number of Robust matches 6274

Number of matches 100177
Number of matches After Lowe's Ratio 26648

80%|██████████ | 32/40 [11:41<02:45, 20.66s/it]

Number of Robust matches 11356

82%|██████████ | 33/40 [12:01<02:24, 20.61s/it]

Number of matches 107031
Number of matches After Lowe's Ratio 13176
Number of Robust matches 6231

85%|██████████ | 34/40 [12:22<02:04, 20.75s/it]

Number of matches 103108
Number of matches After Lowe's Ratio 538
Number of Robust matches 162

88%|██████████ | 35/40 [12:43<01:43, 20.64s/it]

Number of matches 107375
Number of matches After Lowe's Ratio 10398
Number of Robust matches 6021

90%|██████████ | 36/40 [13:04<01:23, 20.91s/it]

Number of matches 108739
Number of matches After Lowe's Ratio 7825
Number of Robust matches 4398

92%|██████████ | 37/40 [13:26<01:03, 21.21s/it]

Number of matches 116973
Number of matches After Lowe's Ratio 5128
Number of Robust matches 3174

95%|██████████ | 38/40 [13:49<00:43, 21.68s/it]

Number of matches 116549
Number of matches After Lowe's Ratio 10440
Number of Robust matches 7613

98%|██████████ | 39/40 [14:12<00:21, 21.86s/it]

Number of matches 107771

Number of matches 107771
Number of matches After Lowe's Ratio 7777
Number of Robust matches 5103

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_right_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 [20]:

```
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_right_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)
```

2%| | 1/61 [00:02<02:16, 2.27s/it]

Number of matches 15850
Number of matches After Lowe's Ratio 1864
Number of Robust matches 892

3%|██████████ | 2/61 [00:04<02:23, 2.43s/it]

Number of matches 20463
Number of matches After Lowe's Ratio 1422
Number of Robust matches 604

5%|██████████ | 3/61 [00:07<02:30, 2.59s/it]

Number of matches 16891
Number of matches After Lowe's Ratio 662
Number of Robust matches 169

7%|██████████ | 4/61 [00:09<02:17, 2.41s/it]

Number of matches 16828
Number of matches After Lowe's Ratio 2820
Number of Robust matches 1597

8%|██████████ | 5/61 [00:12<02:14, 2.40s/it]

Number of matches 17667
Number of matches After Lowe's Ratio 3419
Number of Robust matches 2068

10%|██████████ | 6/61 [00:14<02:14, 2.45s/it]

Number of matches 17727
Number of matches After Lowe's Ratio 3131
Number of Robust matches 1744

11%|██████████ | 7/61 [00:17<02:10, 2.42s/it]

Number of matches 19250
Number of matches After Lowe's Ratio 3535
Number of Robust matches 1843

13%|██████████ | 8/61 [00:19<02:08, 2.42s/it]

Number of matches 12557
Number of matches After Lowe's Ratio 1746
Number of Robust matches 896

15%|██████████ | 9/61 [00:21<01:55, 2.22s/it]

Number of matches 19090
Number of matches After Lowe's Ratio 2766
Number of Robust matches 1684

16%|██████████ | 10/61 [00:23<01:55, 2.27s/it]

Number of matches 12039
Number of matches After Lowe's Ratio 1387
Number of Robust matches 748

18%|██████████ | 11/61 [00:25<01:47, 2.15s/it]

Number of matches 17448
Number of matches After Lowe's Ratio 2917
Number of Robust matches 1573

20%|██████████ | 12/61 [00:27<01:46, 2.18s/it]

Number of matches 15221
Number of matches After Lowe's Ratio 3046
Number of Robust matches 2076

21%|██████████ | 13/61 [00:29<01:42, 2.13s/it]

Number of matches 19009
Number of matches After Lowe's Ratio 3189
Number of Robust matches 2113

23%|██████████ | 14/61 [00:32<01:46, 2.26s/it]

Number of matches 18724
Number of matches After Lowe's Ratio 4514
Number of Robust matches 3453

25%|██████████ | 15/61 [00:35<01:50, 2.40s/it]

Number of matches 18161
Number of matches After Lowe's Ratio 3363
Number of Robust matches 2307

26%|██████████ | 16/61 [00:37<01:50, 2.44s/it]

Number of matches 17507
Number of matches After Lowe's Ratio 3694
Number of Robust matches 2687

28%|██████████ | 17/61 [00:39<01:45, 2.40s/it]

Number of matches 16984
Number of matches After Lowe's Ratio 3422
Number of Robust matches 2732

30%|██████████ | 18/61 [00:42<01:42, 2.38s/it]

Number of matches 16971
Number of matches After Lowe's Ratio 4143
Number of Robust matches 2745

31%|██████████ | 19/61 [00:44<01:41, 2.42s/it]

Number of matches 17121
Number of matches After Lowe's Ratio 4435
Number of Robust matches 3506

33%|██████████ | 20/61 [00:47<01:43, 2.54s/it]

Number of matches 17331
Number of matches After Lowe's Ratio 3623
Number of Robust matches 2797

34%|██████ | 21/61 [00:49<01:38, 2.46s/it]

Number of matches 19219
Number of matches After Lowe's Ratio 3262
Number of Robust matches 2193

36%|██████ | 22/61 [00:52<01:37, 2.50s/it]

Number of matches 18480
Number of matches After Lowe's Ratio 3262
Number of Robust matches 2069

38%|██████ | 23/61 [00:54<01:35, 2.51s/it]

Number of matches 19423
Number of matches After Lowe's Ratio 3643
Number of Robust matches 2352

39%|██████ | 24/61 [00:57<01:38, 2.66s/it]

Number of matches 19540
Number of matches After Lowe's Ratio 2894
Number of Robust matches 1863

41%|██████ | 25/61 [01:00<01:36, 2.68s/it]

Number of matches 23070
Number of matches After Lowe's Ratio 2662
Number of Robust matches 1292

43%|██████ | 26/61 [01:03<01:39, 2.83s/it]

Number of matches 19327
Number of matches After Lowe's Ratio 2823
Number of Robust matches 1499

44%|██████ | 27/61 [01:06<01:35, 2.80s/it]

Number of matches 21616
Number of matches After Lowe's Ratio 2834
Number of Robust matches 1436

46%|██████ | 28/61 [01:09<01:36, 2.91s/it]

Number of matches 19935
Number of matches After Lowe's Ratio 2820
Number of Robust matches 1127

48%|██████ | 29/61 [01:12<01:32, 2.90s/it]

Number of matches 22791
Number of matches After Lowe's Ratio 1780
Number of Robust matches 757

49%|██████ | 30/61 [01:15<01:32, 2.98s/it]

Number of matches 21497
Number of matches After Lowe's Ratio 2667
Number of Robust matches 1415

51%|██████ | 31/61 [01:19<01:35, 3.19s/it]

Number of matches 20351
Number of matches After Lowe's Ratio 1310
Number of Robust matches 559

52%|██████ | 32/61 [01:22<01:27, 3.02s/it]

Number of matches 17412
Number of matches After Lowe's Ratio 929
Number of Robust matches 261

54%|██████ | 33/61 [01:24<01:17, 2.78s/it]

Number of matches 16896
Number of matches After Lowe's Ratio 2257
Number of Robust matches 1236

56%|██████ | 34/61 [01:26<01:10, 2.62s/it]

Number of matches 16303
Number of matches After Lowe's Ratio 2571
Number of Robust matches 1471

57%|██████ | 35/61 [01:28<01:05, 2.51s/it]

Number of matches 18249
Number of matches After Lowe's Ratio 2252
Number of Robust matches 1292

59%|██████ | 36/61 [01:31<01:04, 2.58s/it]

Number of matches 21853
Number of matches After Lowe's Ratio 3022
Number of Robust matches 1644

61%|██████ | 37/61 [01:34<01:05, 2.74s/it]

Number of matches 24851
Number of matches After Lowe's Ratio 2882
Number of Robust matches 1244

62%|██████ | 38/61 [01:38<01:11, 3.10s/it]

Number of matches 28347
Number of matches After Lowe's Ratio 3345
Number of Robust matches 1259

64%|██████ | 39/61 [01:43<01:18, 3.56s/it]

Number of matches 24822
Number of matches After Lowe's Ratio 3145
Number of Robust matches 1417

66%|██████ | 40/61 [01:46<01:14, 3.55s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 3165
Number of Robust matches 1640

67%|██████ | 41/61 [01:49<01:05, 3.28s/it]

Number of matches 18074
Number of matches After Lowe's Ratio 3264
Number of Robust matches 1942

69%|██████ | 42/61 [01:52<01:01, 3.25s/it]

Number of matches 16132
Number of matches After Lowe's Ratio 3456
Number of Robust matches 2218

70%|██████ | 43/61 [01:54<00:52, 2.93s/it]

Number of matches 16505
Number of matches After Lowe's Ratio 3773
Number of Robust matches 2617

72%|██████ | 44/61 [01:56<00:45, 2.70s/it]

Number of matches 17795
Number of matches After Lowe's Ratio 3412
Number of Robust matches 2404

74%|██████ | 45/61 [01:59<00:41, 2.60s/it]

Number of matches 19052
Number of matches After Lowe's Ratio 3898
Number of Robust matches 2417

75%|██████ | 46/61 [02:01<00:38, 2.58s/it]

Number of matches 18726
Number of matches After Lowe's Ratio 4306
Number of Robust matches 2332

77%|██████ | 47/61 [02:04<00:37, 2.64s/it]

Number of matches 18580
Number of matches After Lowe's Ratio 4301
Number of Robust matches 2603

79%|██████ | 48/61 [02:07<00:34, 2.65s/it]

Number of matches 15741
Number of matches After Lowe's Ratio 2721
Number of Robust matches 1778

80%|██████ | 49/61 [02:09<00:29, 2.46s/it]

Number of matches 14586
Number of matches After Lowe's Ratio 4016
Number of Robust matches 3026

82% | ████████ | 50/61 [02:11<00:25, 2.29s/it]

Number of matches 16381
Number of matches After Lowe's Ratio 3871
Number of Robust matches 2917

84% | ████████ | 51/61 [02:13<00:22, 2.24s/it]

Number of matches 15190
Number of matches After Lowe's Ratio 2743
Number of Robust matches 1980

85% | ████████ | 52/61 [02:15<00:20, 2.26s/it]

Number of matches 16204
Number of matches After Lowe's Ratio 2838
Number of Robust matches 1911

87% | ████████ | 53/61 [02:17<00:17, 2.23s/it]

Number of matches 16360
Number of matches After Lowe's Ratio 3636
Number of Robust matches 2676

89% | ████████ | 54/61 [02:19<00:15, 2.21s/it]

Number of matches 16749
Number of matches After Lowe's Ratio 3021
Number of Robust matches 2050

90% | ████████ | 55/61 [02:22<00:13, 2.25s/it]

Number of matches 16958
Number of matches After Lowe's Ratio 3343
Number of Robust matches 2351

92% | ████████ | 56/61 [02:24<00:11, 2.24s/it]

Number of matches 16883
Number of matches After Lowe's Ratio 2923
Number of Robust matches 1725

93% | ████████ | 57/61 [02:27<00:09, 2.46s/it]

Number of matches 16697
Number of matches After Lowe's Ratio 4146
Number of Robust matches 2326

95% | ████████ | 58/61 [02:29<00:07, 2.36s/it]

Number of matches 17245
Number of matches After Lowe's Ratio 2536
Number of Robust matches 1149

Number of Robust matches 1119

97% | ██████████ | 59/61 [02:31<00:04, 2.36s/it]

Number of matches 16937
Number of matches After Lowe's Ratio 3663
Number of Robust matches 1480

98% | ██████████ | 60/61 [02:34<00:02, 2.57s/it]
0% | ██████████ | 0/40 [00:00<?, ?it/s]

Number of matches 14790
Number of matches After Lowe's Ratio 1402
Number of Robust matches 499

2% | ██████████ | 1/40 [00:02<01:36, 2.47s/it]

Number of matches 17910
Number of matches After Lowe's Ratio 1854
Number of Robust matches 962

5% | ██████████ | 2/40 [00:04<01:33, 2.45s/it]

Number of matches 20488
Number of matches After Lowe's Ratio 2848
Number of Robust matches 1975

8% | ██████████ | 3/40 [00:07<01:34, 2.57s/it]

Number of matches 14865
Number of matches After Lowe's Ratio 2593
Number of Robust matches 1700

10% | ██████████ | 4/40 [00:09<01:21, 2.25s/it]

Number of matches 10652
Number of matches After Lowe's Ratio 1446
Number of Robust matches 714

12% | ██████████ | 5/40 [00:10<01:07, 1.93s/it]

Number of matches 14443
Number of matches After Lowe's Ratio 1095
Number of Robust matches 473

15% | ██████████ | 6/40 [00:12<01:03, 1.86s/it]

Number of matches 10456
Number of matches After Lowe's Ratio 2231
Number of Robust matches 1543

18% | ██████████ | 7/40 [00:14<01:00, 1.83s/it]

Number of matches 17715
Number of matches After Lowe's Ratio 1428
Number of Robust matches 857

20%|██████ | 8/40 [00:16<01:03, 1.99s/it]

Number of matches 18284
Number of matches After Lowe's Ratio 4210
Number of Robust matches 2756

22%|██████ | 9/40 [00:18<01:05, 2.10s/it]

Number of matches 17764
Number of matches After Lowe's Ratio 4283
Number of Robust matches 3332

25%|██████ | 10/40 [00:21<01:05, 2.18s/it]

Number of matches 17499
Number of matches After Lowe's Ratio 3700
Number of Robust matches 2799

28%|██████ | 11/40 [00:23<01:06, 2.31s/it]

Number of matches 19138
Number of matches After Lowe's Ratio 3608
Number of Robust matches 2533

30%|██████ | 12/40 [00:27<01:12, 2.59s/it]

Number of matches 21978
Number of matches After Lowe's Ratio 2807
Number of Robust matches 1824

32%|██████ | 13/40 [00:30<01:14, 2.74s/it]

Number of matches 23315
Number of matches After Lowe's Ratio 3682
Number of Robust matches 2372

35%|██████ | 14/40 [00:33<01:16, 2.94s/it]

Number of matches 25930
Number of matches After Lowe's Ratio 3593
Number of Robust matches 2111

38%|██████ | 15/40 [00:37<01:23, 3.33s/it]

Number of matches 25725
Number of matches After Lowe's Ratio 4076
Number of Robust matches 1883

40%|██████ | 16/40 [00:41<01:22, 3.45s/it]

Number of matches 25272
Number of matches After Lowe's Ratio 4223
Number of Robust matches 2001

42%|██████ | 17/40 [00:45<01:20, 3.50s/it]

Number of matches 23716

Number of matches After Lowe's Ratio 4358
Number of Robust matches 1944

45% | ████████ | 18/40 [00:48<01:18, 3.57s/it]

Number of matches 21541
Number of matches After Lowe's Ratio 3406
Number of Robust matches 1586

48% | ████████ | 19/40 [00:51<01:11, 3.39s/it]

Number of matches 20126
Number of matches After Lowe's Ratio 4084
Number of Robust matches 1736

50% | ████████ | 20/40 [00:54<01:04, 3.22s/it]

Number of matches 18854
Number of matches After Lowe's Ratio 3356
Number of Robust matches 1321

52% | ████████ | 21/40 [00:58<01:01, 3.25s/it]

Number of matches 17303
Number of matches After Lowe's Ratio 3023
Number of Robust matches 1290

55% | ████████ | 22/40 [01:00<00:53, 2.98s/it]

Number of matches 18642
Number of matches After Lowe's Ratio 2499
Number of Robust matches 1178

57% | ████████ | 23/40 [01:03<00:49, 2.93s/it]

Number of matches 27086
Number of matches After Lowe's Ratio 1016
Number of Robust matches 330

60% | ████████ | 24/40 [01:07<00:51, 3.23s/it]

Number of matches 22491
Number of matches After Lowe's Ratio 1638
Number of Robust matches 670

62% | ████████ | 25/40 [01:10<00:49, 3.32s/it]

Number of matches 31012
Number of matches After Lowe's Ratio 465
Number of Robust matches 16

65% | ████████ | 26/40 [01:15<00:51, 3.69s/it]

Number of matches 24213
Number of matches After Lowe's Ratio 1767
Number of Robust matches 547

68%|██████████ | 27/40 [01:18<00:47, 3.68s/it]

Number of matches 22667
Number of matches After Lowe's Ratio 3411
Number of Robust matches 1349

70%|██████████ | 28/40 [01:22<00:42, 3.54s/it]

Number of matches 19376
Number of matches After Lowe's Ratio 2831
Number of Robust matches 1159

72%|██████████ | 29/40 [01:24<00:35, 3.26s/it]

Number of matches 18221
Number of matches After Lowe's Ratio 2553
Number of Robust matches 1072

75%|██████████ | 30/40 [01:27<00:30, 3.03s/it]

Number of matches 19609
Number of matches After Lowe's Ratio 2916
Number of Robust matches 1020

78%|██████████ | 31/40 [01:30<00:28, 3.12s/it]

Number of matches 19236
Number of matches After Lowe's Ratio 2722
Number of Robust matches 1027

80%|██████████ | 32/40 [01:33<00:24, 3.03s/it]

Number of matches 18754
Number of matches After Lowe's Ratio 4227
Number of Robust matches 1692

82%|██████████ | 33/40 [01:35<00:20, 2.89s/it]

Number of matches 20522
Number of matches After Lowe's Ratio 2758
Number of Robust matches 972

85%|██████████ | 34/40 [01:38<00:17, 2.87s/it]

Number of matches 20368
Number of matches After Lowe's Ratio 4274
Number of Robust matches 1920

88%|██████████ | 35/40 [01:41<00:14, 2.98s/it]

Number of matches 19692
Number of matches After Lowe's Ratio 3196
Number of Robust matches 1570

90%|██████████ | 36/40 [01:44<00:11, 2.87s/it]

Number of matches 17996

Number of matches After Lowe's Ratio 2538
Number of Robust matches 1315

92% | ██████████ | 37/40 [01:46<00:08, 2.71s/it]

Number of matches 17038
Number of matches After Lowe's Ratio 2158
Number of Robust matches 1567

95% | ██████████ | 38/40 [01:49<00:05, 2.55s/it]

Number of matches 17238
Number of matches After Lowe's Ratio 2579
Number of Robust matches 1636

98% | ██████████ | 39/40 [01:51<00:02, 2.85s/it]

Number of matches 16004
Number of matches After Lowe's Ratio 2294
Number of Robust matches 1519

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_right_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 [20]:

```
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_surfsift[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_right_surfsift[j:j+2][::-1],points_all_right_surfsift[j:j+2][::-1],descriptors_all_right_surfsift[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 [00:05<05:02, 5.04s/it]

Number of matches 29759
Number of matches After Lowe's Ratio 3234
Number of Robust matches 1043

3%| | 2/61 [00:10<05:07, 5.21s/it]

Number of matches 31803
Number of matches After Lowe's Ratio 2860
Number of Robust matches 832

5%| | 3/61 [00:15<05:11, 5.36s/it]

Number of matches 28470
Number of matches After Lowe's Ratio 620
Number of Robust matches 143

7%| | 4/61 [00:20<04:58, 5.24s/it]

Number of matches 30574
Number of matches After Lowe's Ratio 6266
Number of Robust matches 2236

8%| | 5/61 [00:26<04:50, 5.18s/it]

Number of matches 28292
Number of matches After Lowe's Ratio 5580
Number of Robust matches 2092

10%| | 6/61 [00:30<04:37, 5.05s/it]

Number of matches 28159
Number of matches After Lowe's Ratio 5507
Number of Robust matches 2039

11%| | 7/61 [00:35<04:28, 4.96s/it]

Number of matches 30159
Number of matches After Lowe's Ratio 5929
Number of Robust matches 2156

13%| | 8/61 [00:40<04:20, 4.91s/it]

Number of matches 25112
Number of matches After Lowe's Ratio 2905
Number of Robust matches 1226

15%|██████████ | 9/61 [00:44<04:08, 4.77s/it]

Number of matches 29686
Number of matches After Lowe's Ratio 5277
Number of Robust matches 2265

16%|██████████ | 10/61 [00:50<04:09, 4.89s/it]

Number of matches 23395
Number of matches After Lowe's Ratio 2677
Number of Robust matches 1038

18%|██████████ | 11/61 [00:54<03:54, 4.69s/it]

Number of matches 27431
Number of matches After Lowe's Ratio 5847
Number of Robust matches 2953

20%|██████████ | 12/61 [00:58<03:44, 4.58s/it]

Number of matches 27034
Number of matches After Lowe's Ratio 5789
Number of Robust matches 2669

21%|██████████ | 13/61 [01:03<03:38, 4.55s/it]

Number of matches 29867
Number of matches After Lowe's Ratio 5503
Number of Robust matches 2716

23%|██████████ | 14/61 [01:08<03:45, 4.80s/it]

Number of matches 31522
Number of matches After Lowe's Ratio 9201
Number of Robust matches 5247

25%|██████████ | 15/61 [01:13<03:46, 4.92s/it]

Number of matches 30097
Number of matches After Lowe's Ratio 6944
Number of Robust matches 3934

26%|██████████ | 16/61 [01:19<03:50, 5.13s/it]

Number of matches 30110
Number of matches After Lowe's Ratio 7524
Number of Robust matches 4196

28%|██████████ | 17/61 [01:24<03:45, 5.13s/it]

Number of matches 29782
Number of matches After Lowe's Ratio 7370
Number of Robust matches 4517

30%|██████ | 18/61 [01:29<03:41, 5.16s/it]

Number of matches 29162
Number of matches After Lowe's Ratio 7743
Number of Robust matches 4434

31%|██████ | 19/61 [01:34<03:33, 5.09s/it]

Number of matches 28941
Number of matches After Lowe's Ratio 8975
Number of Robust matches 5190

33%|██████ | 20/61 [01:39<03:27, 5.07s/it]

Number of matches 29718
Number of matches After Lowe's Ratio 7640
Number of Robust matches 4165

34%|██████ | 21/61 [01:44<03:21, 5.04s/it]

Number of matches 29774
Number of matches After Lowe's Ratio 6859
Number of Robust matches 3287

36%|██████ | 22/61 [01:49<03:20, 5.13s/it]

Number of matches 28465
Number of matches After Lowe's Ratio 7437
Number of Robust matches 3964

38%|██████ | 23/61 [01:55<03:14, 5.13s/it]

Number of matches 28901
Number of matches After Lowe's Ratio 6904
Number of Robust matches 3177

39%|██████ | 24/61 [01:59<03:08, 5.09s/it]

Number of matches 29317
Number of matches After Lowe's Ratio 6494
Number of Robust matches 3115

41%|██████ | 25/61 [02:04<02:59, 4.99s/it]

Number of matches 32274
Number of matches After Lowe's Ratio 4300
Number of Robust matches 1650

43%|██████ | 26/61 [02:10<02:59, 5.12s/it]

Number of matches 29333
Number of matches After Lowe's Ratio 4939
Number of Robust matches 1701

44%|██████ | 27/61 [02:15<02:51, 5.06s/it]

Number of matches 28791
Number of matches After Lowe's Ratio 5713
Number of Robust matches 2072

46% | [REDACTED] | 28/61 [02:19<02:44, 4.98s/it]

Number of matches 31214
Number of matches After Lowe's Ratio 5776
Number of Robust matches 1440

48% | [REDACTED] | 29/61 [02:25<02:47, 5.23s/it]

Number of matches 28464
Number of matches After Lowe's Ratio 3290
Number of Robust matches 972

49% | [REDACTED] | 30/61 [02:30<02:37, 5.08s/it]

Number of matches 29409
Number of matches After Lowe's Ratio 5220
Number of Robust matches 1827

51% | [REDACTED] | 31/61 [02:35<02:31, 5.04s/it]

Number of matches 27445
Number of matches After Lowe's Ratio 2586
Number of Robust matches 953

52% | [REDACTED] | 32/61 [02:39<02:20, 4.84s/it]

Number of matches 26763
Number of matches After Lowe's Ratio 1057
Number of Robust matches 241

54% | [REDACTED] | 33/61 [02:44<02:14, 4.80s/it]

Number of matches 27227
Number of matches After Lowe's Ratio 5150
Number of Robust matches 2390

56% | [REDACTED] | 34/61 [02:48<02:05, 4.66s/it]

Number of matches 26057
Number of matches After Lowe's Ratio 6232
Number of Robust matches 2546

57% | [REDACTED] | 35/61 [02:52<01:56, 4.49s/it]

Number of matches 27493
Number of matches After Lowe's Ratio 5444
Number of Robust matches 2498

59% | [REDACTED] | 36/61 [02:58<01:58, 4.76s/it]

Number of matches 28655
Number of matches After Lowe's Ratio 6483
Number of Robust matches 2746

61% | ████████ | 37/61 [03:02<01:54, 4.75s/it]

Number of matches 31224
Number of matches After Lowe's Ratio 4773
Number of Robust matches 1597

62% | ████████ | 38/61 [03:08<01:54, 4.98s/it]

Number of matches 32243
Number of matches After Lowe's Ratio 5610
Number of Robust matches 1574

64% | ████████ | 39/61 [03:13<01:52, 5.11s/it]

Number of matches 31404
Number of matches After Lowe's Ratio 5038
Number of Robust matches 1778

66% | ████████ | 40/61 [03:19<01:48, 5.19s/it]

Number of matches 29941
Number of matches After Lowe's Ratio 6495
Number of Robust matches 2987

67% | ████████ | 41/61 [03:24<01:42, 5.13s/it]

Number of matches 29731
Number of matches After Lowe's Ratio 6903
Number of Robust matches 3171

69% | ████████ | 42/61 [03:29<01:39, 5.25s/it]

Number of matches 29003
Number of matches After Lowe's Ratio 7143
Number of Robust matches 4138

70% | ████████ | 43/61 [03:34<01:31, 5.09s/it]

Number of matches 29540
Number of matches After Lowe's Ratio 8508
Number of Robust matches 4669

72% | ████████ | 44/61 [03:39<01:27, 5.12s/it]

Number of matches 30714
Number of matches After Lowe's Ratio 7699
Number of Robust matches 4092

74% | ████████ | 45/61 [03:44<01:21, 5.11s/it]

Number of matches 31999
Number of matches After Lowe's Ratio 8189
Number of Robust matches 4288

75%|██████████ | 46/61 [03:50<01:18, 5.26s/it]

Number of matches 31349
Number of matches After Lowe's Ratio 8440
Number of Robust matches 4366

77%|██████████ | 47/61 [03:55<01:14, 5.29s/it]

Number of matches 31491
Number of matches After Lowe's Ratio 7710
Number of Robust matches 3627

79%|██████████ | 48/61 [04:01<01:11, 5.49s/it]

Number of matches 29793
Number of matches After Lowe's Ratio 4667
Number of Robust matches 2204

80%|██████████ | 49/61 [04:06<01:03, 5.30s/it]

Number of matches 28686
Number of matches After Lowe's Ratio 8551
Number of Robust matches 4835

82%|██████████ | 50/61 [04:11<00:57, 5.24s/it]

Number of matches 29518
Number of matches After Lowe's Ratio 8930
Number of Robust matches 4928

84%|██████████ | 51/61 [04:16<00:50, 5.09s/it]

Number of matches 27220
Number of matches After Lowe's Ratio 5383
Number of Robust matches 2976

85%|██████████ | 52/61 [04:20<00:44, 4.89s/it]

Number of matches 27631
Number of matches After Lowe's Ratio 5593
Number of Robust matches 2417

87%|██████████ | 53/61 [04:25<00:38, 4.82s/it]

Number of matches 28608
Number of matches After Lowe's Ratio 7346
Number of Robust matches 3505

89%|██████████ | 54/61 [04:30<00:33, 4.74s/it]

Number of matches 27793
Number of matches After Lowe's Ratio 5319
Number of Robust matches 2427

90%|██████████ | 55/61 [04:34<00:28, 4.76s/it]

Number of matches 26647
Number of matches After Lowe's Ratio 5656

Number of Robust matches 2698

92%|██████████| 56/61 [04:39<00:23, 4.74s/it]

Number of matches 27094
Number of matches After Lowe's Ratio 5035
Number of Robust matches 2199

93%|██████████| 57/61 [04:44<00:18, 4.68s/it]

Number of matches 28742
Number of matches After Lowe's Ratio 7467
Number of Robust matches 2932

95%|██████████| 58/61 [04:48<00:14, 4.71s/it]

Number of matches 28988
Number of matches After Lowe's Ratio 4167
Number of Robust matches 1183

97%|██████████| 59/61 [04:53<00:09, 4.71s/it]

Number of matches 29103
Number of matches After Lowe's Ratio 6528
Number of Robust matches 1829

98%|██████████| 60/61 [04:58<00:04, 4.98s/it]
0%|██████████| 0/40 [00:00<?, ?it/s]

Number of matches 28791
Number of matches After Lowe's Ratio 1698
Number of Robust matches 414

2%|██████████| 1/40 [00:05<03:18, 5.09s/it]

Number of matches 29394
Number of matches After Lowe's Ratio 3489
Number of Robust matches 1651

5%|██████████| 2/40 [00:10<03:16, 5.16s/it]

Number of matches 30994
Number of matches After Lowe's Ratio 5695
Number of Robust matches 3244

8%|██████████| 3/40 [00:15<03:11, 5.17s/it]

Number of matches 27439
Number of matches After Lowe's Ratio 5502
Number of Robust matches 3138

10%|██████████| 4/40 [00:19<02:55, 4.88s/it]

Number of matches 22006
Number of matches After Lowe's Ratio 2736
Number of Robust matches 1407

12%|██████████ | 5/40 [00:23<02:35, 4.44s/it]

Number of matches 25745
Number of matches After Lowe's Ratio 1682
Number of Robust matches 744

15%|██████████ | 6/40 [00:27<02:26, 4.30s/it]

Number of matches 23382
Number of matches After Lowe's Ratio 5838
Number of Robust matches 3039

18%|██████████ | 7/40 [00:31<02:20, 4.27s/it]

Number of matches 32804
Number of matches After Lowe's Ratio 3423
Number of Robust matches 1606

20%|██████████ | 8/40 [00:37<02:31, 4.74s/it]

Number of matches 30475
Number of matches After Lowe's Ratio 10511
Number of Robust matches 5862

22%|██████████ | 9/40 [00:43<02:34, 4.98s/it]

Number of matches 32449
Number of matches After Lowe's Ratio 10521
Number of Robust matches 5982

25%|██████████ | 10/40 [00:49<02:38, 5.28s/it]

Number of matches 28137
Number of matches After Lowe's Ratio 7809
Number of Robust matches 4959

28%|██████████ | 11/40 [00:53<02:30, 5.18s/it]

Number of matches 28712
Number of matches After Lowe's Ratio 8483
Number of Robust matches 5181

30%|██████████ | 12/40 [00:58<02:21, 5.04s/it]

Number of matches 27853
Number of matches After Lowe's Ratio 5998
Number of Robust matches 3348

32%|██████████ | 13/40 [01:03<02:16, 5.06s/it]

Number of matches 29532
Number of matches After Lowe's Ratio 7929
Number of Robust matches 3453

35%|██████████ | 14/40 [01:08<02:10, 5.03s/it]

Number of matches 30289

Number of matches 30209
Number of matches After Lowe's Ratio 7660
Number of Robust matches 3719

38% | ████████ | 15/40 [01:14<02:09, 5.16s/it]

Number of matches 31517
Number of matches After Lowe's Ratio 8533
Number of Robust matches 3281

40% | ████████ | 16/40 [01:19<02:07, 5.33s/it]

Number of matches 30146
Number of matches After Lowe's Ratio 7763
Number of Robust matches 3706

42% | ████████ | 17/40 [01:25<02:02, 5.32s/it]

Number of matches 31185
Number of matches After Lowe's Ratio 8895
Number of Robust matches 3425

45% | ████████ | 18/40 [01:30<01:58, 5.37s/it]

Number of matches 32266
Number of matches After Lowe's Ratio 7897
Number of Robust matches 2855

48% | ████████ | 19/40 [01:36<01:55, 5.48s/it]

Number of matches 31322
Number of matches After Lowe's Ratio 9637
Number of Robust matches 3324

50% | ████████ | 20/40 [01:41<01:48, 5.41s/it]

Number of matches 27735
Number of matches After Lowe's Ratio 7486
Number of Robust matches 2819

52% | ████████ | 21/40 [01:46<01:39, 5.23s/it]

Number of matches 30089
Number of matches After Lowe's Ratio 7567
Number of Robust matches 2134

55% | ████████ | 22/40 [01:51<01:35, 5.29s/it]

Number of matches 27897
Number of matches After Lowe's Ratio 5985
Number of Robust matches 2574

57% | ████████ | 23/40 [01:56<01:28, 5.21s/it]

Number of matches 32011
Number of matches After Lowe's Ratio 1826
Number of Robust matches 572

60%|██████ | 24/40 [02:02<01:24, 5.25s/it]

Number of matches 30610
Number of matches After Lowe's Ratio 3963
Number of Robust matches 1167

62%|██████ | 25/40 [02:07<01:18, 5.24s/it]

Number of matches 34036
Number of matches After Lowe's Ratio 62
Number of Robust matches 10

65%|██████ | 26/40 [02:13<01:16, 5.44s/it]

Number of matches 31567
Number of matches After Lowe's Ratio 3525
Number of Robust matches 1104

68%|██████ | 27/40 [02:18<01:10, 5.42s/it]

Number of matches 29422
Number of matches After Lowe's Ratio 7215
Number of Robust matches 2748

70%|██████ | 28/40 [02:24<01:04, 5.38s/it]

Number of matches 28350
Number of matches After Lowe's Ratio 6040
Number of Robust matches 2246

72%|██████ | 29/40 [02:28<00:56, 5.17s/it]

Number of matches 28706
Number of matches After Lowe's Ratio 5787
Number of Robust matches 1743

75%|██████ | 30/40 [02:33<00:51, 5.15s/it]

Number of matches 32873
Number of matches After Lowe's Ratio 5377
Number of Robust matches 1454

78%|██████ | 31/40 [02:39<00:48, 5.35s/it]

Number of matches 33122
Number of matches After Lowe's Ratio 5766
Number of Robust matches 1680

80%|██████ | 32/40 [02:45<00:43, 5.49s/it]

Number of matches 32633
Number of matches After Lowe's Ratio 9346
Number of Robust matches 2945

82%|██████ | 33/40 [02:51<00:38, 5.51s/it]

Number of matches 29972
Number of matches After Lowe's Ratio 5324
Number of Robust matches 1959

85% | ██████████ | 34/40 [02:56<00:33, 5.58s/it]

Number of matches 27587
Number of matches After Lowe's Ratio 7314
Number of Robust matches 2687

88% | ██████████ | 35/40 [03:01<00:26, 5.25s/it]

Number of matches 28008
Number of matches After Lowe's Ratio 6435
Number of Robust matches 2840

90% | ██████████ | 36/40 [03:06<00:20, 5.13s/it]

Number of matches 26930
Number of matches After Lowe's Ratio 5042
Number of Robust matches 1958

92% | ██████████ | 37/40 [03:10<00:14, 4.93s/it]

Number of matches 26784
Number of matches After Lowe's Ratio 3641
Number of Robust matches 2212

95% | ██████████ | 38/40 [03:15<00:09, 4.87s/it]

Number of matches 27835
Number of matches After Lowe's Ratio 5351
Number of Robust matches 2808

98% | ██████████ | 39/40 [03:20<00:05, 5.13s/it]

Number of matches 26661
Number of matches After Lowe's Ratio 5350
Number of Robust matches 2678

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_right_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 [21]:

```

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 [22]:

```

def final_steps_left_union(images_left,H_left,xmax,xmin,ymax,ymin,t,h,w,Ht):

```

```

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]:

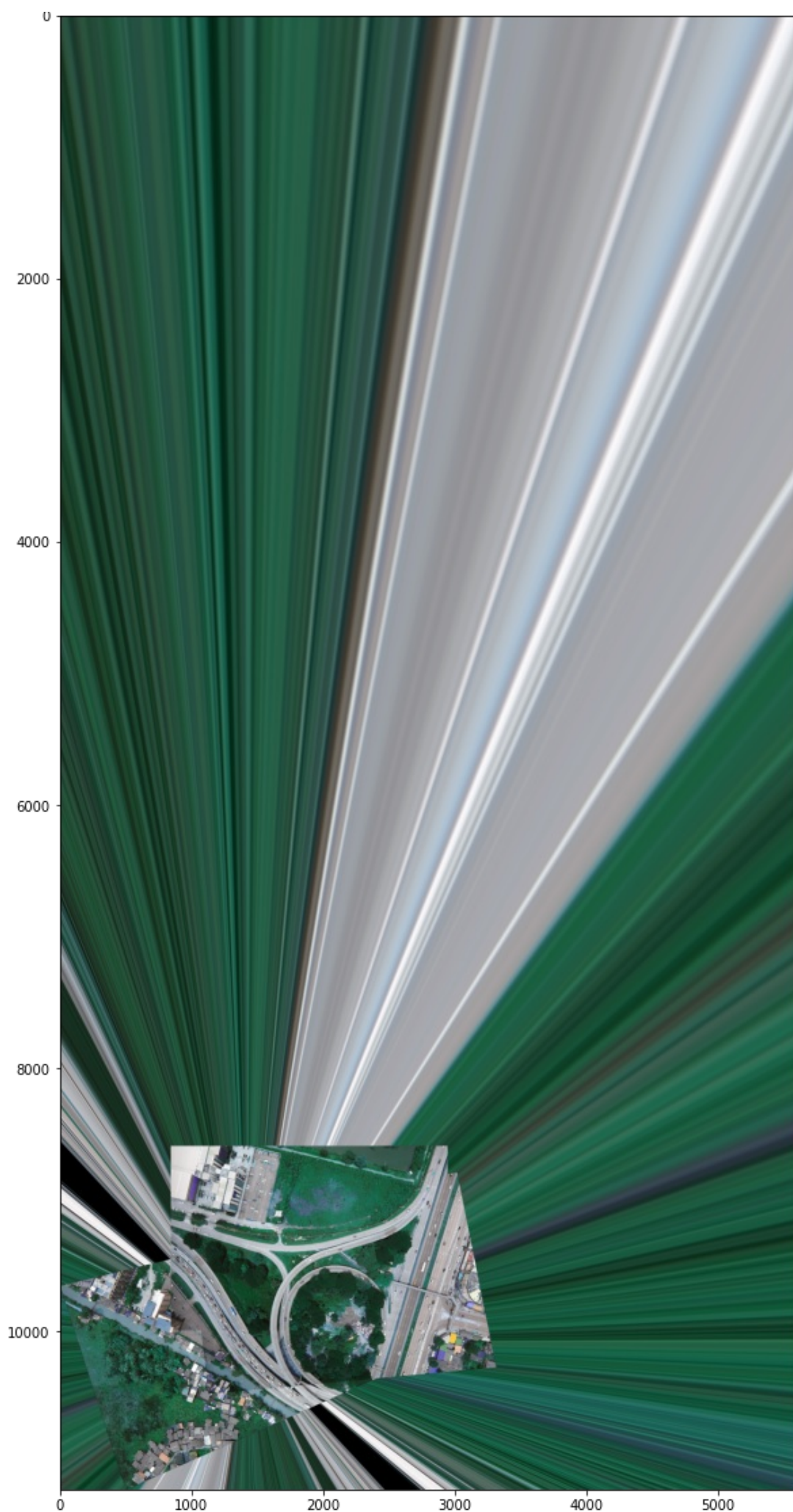
```

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 [23]:

```
xmax,xmin,ymax,ymin,t,h,w,Ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_no_enhance,H_left_surfsift,H_right_surfsift)
```

Step1:Done
Step2:Done

In []:

```
warp_imgs_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_surfsift,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

In []:

```
warp_imgs_all_surfsift = final_step_right_union(warp_imgs_left,images_right_bgr_no_enhance,H_right_star,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

In []:

```
plt.figure(figsize=(20,20))  
plt.imshow(warp_imgs_all_surfsift)  
plt.title(' 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_sift,H_right_sift)
```

Step1:Done
Step2:Done

In [24]:

```
warp_img = final_steps_left_union(images_left_bgr_no_enhance,H_left_sift,omax,omin,umax,umin,T,H,W,HT)
```

step31:Done

In [25]:

```
warp_imgs_all_sift = final_step_right_union(warp_img,images_right_bgr_no_enhance,H_right_sift,omax,omin,umax,umin,T,H,W,HT)
```

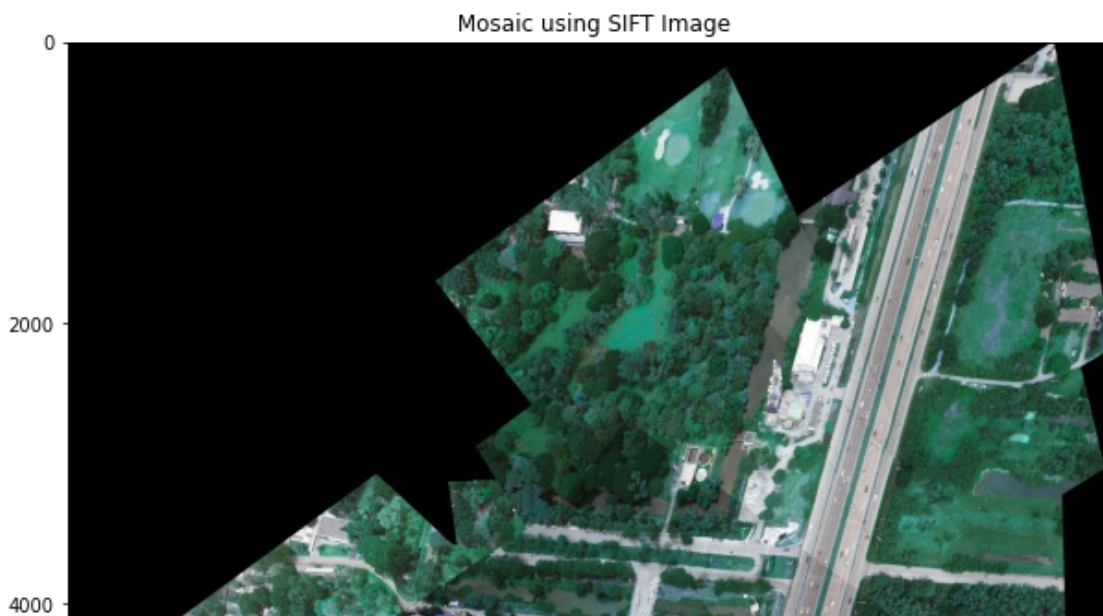
step32:Done

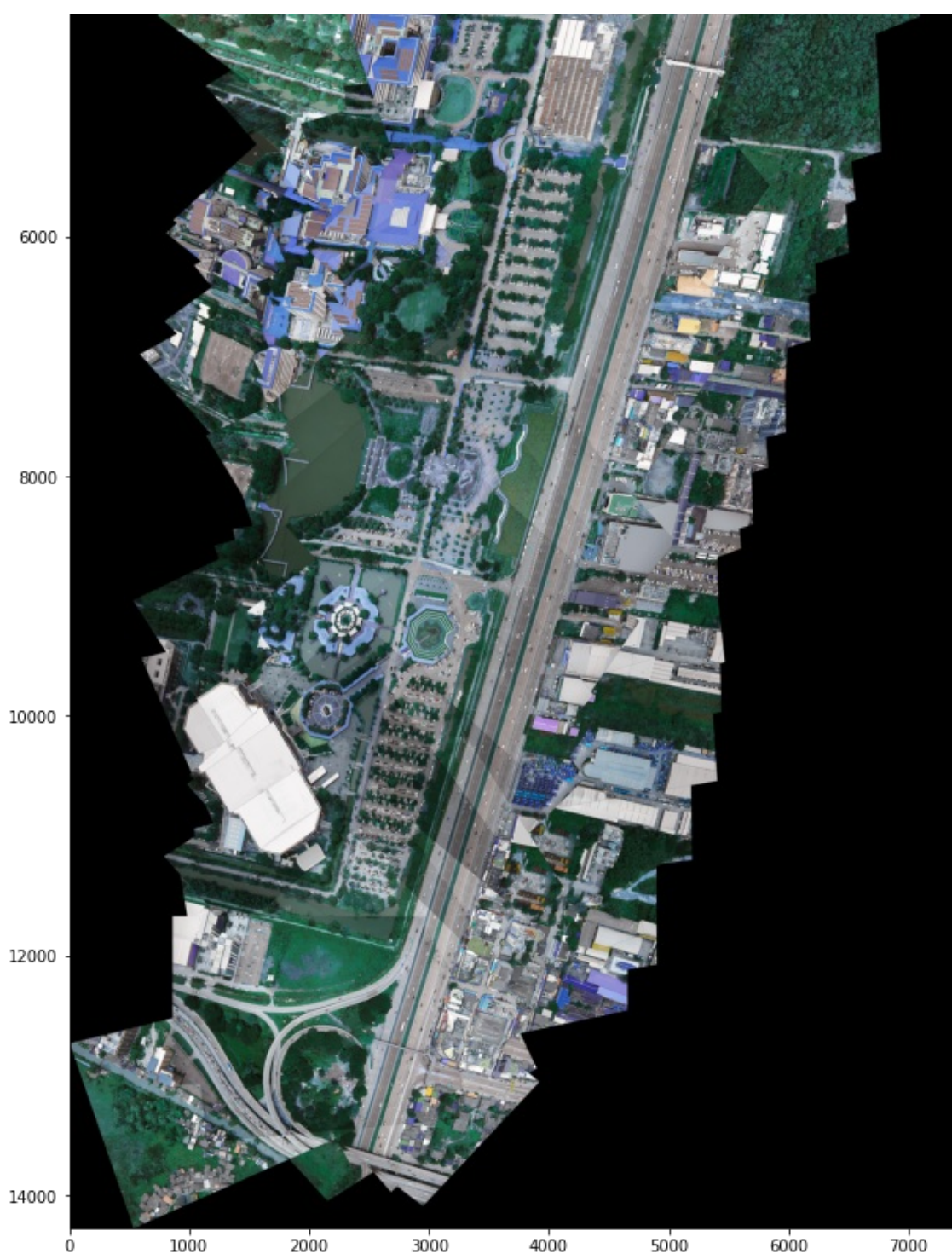
In [26]:

```
plt.figure(figsize=(20,20))  
plt.imshow(warp_imgs_all_sift)  
plt.title(' Mosaic using SIFT Image')
```

Out[26]:

Text(0.5, 1.0, ' Mosaic using SIFT Image')





In [23]:

```
mmax,mmin,nmax,nmin,d,e,f,g = warpnImages(images_left_bgr_no_enhance, images_right_bgr_no_enhance,H_left_fast,H_right_fast)
```

Step1:Done

Step2:Done

In [24]:

```
warp_imgs_fast = final_steps_left_union(images_left_bgr_no_enhance,H_left_fast,mmax,mmin,nmax,nmin,d,e,f,g)
```

step31:Done

In [25]:

```
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)
```

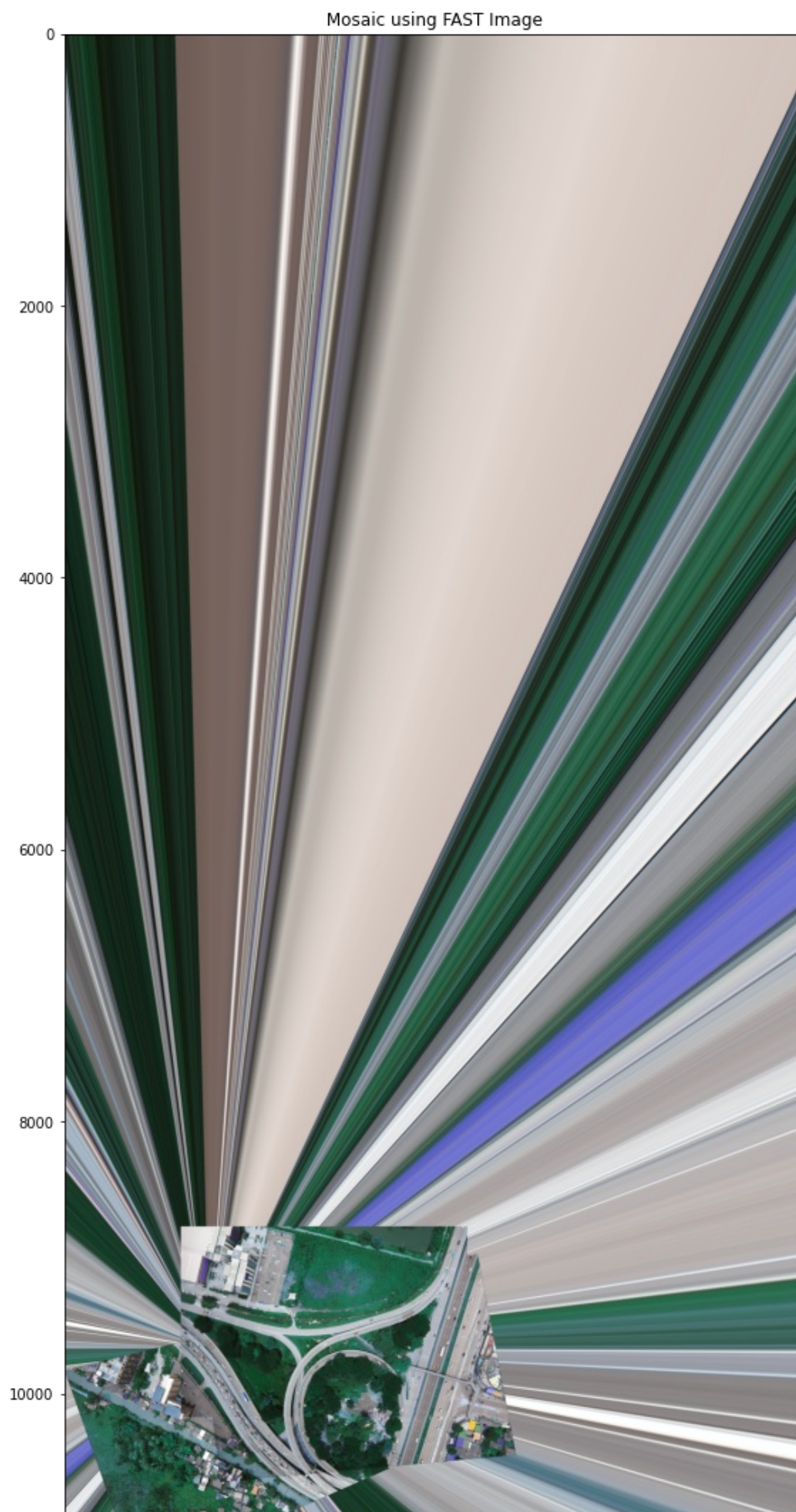
step32:Done

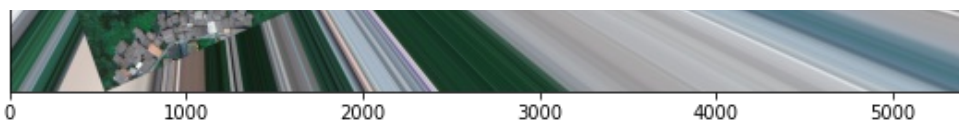
In [26]:

```
plt.figure(figsize=(20,20))  
plt.imshow(warp_imgs_all_fast)  
plt.title(' Mosaic using FAST Image')
```

Out[26]:

```
Text(0.5, 1.0, ' 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, amin, 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, 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_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 [ ]:
```