In [20]:

```
!pip install torchsummary
```

```
Requirement already satisfied: torchsummary in /opt/conda/lib/python3.7/site-packages (1.
5.1)
```

In [21]:

```
import numpy as np
import cv2
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 [22]:

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

inliner_matchset = []
def features_matching(a,keypointlength,threshold):
    bestmatch = np.empty((keypointlength), dtype=np.int16)
    img1index = np.empty((keypointlength),dtype=np.init16)
    distance = np.empty((keypointlength))
    index =0
    for j in range(0,keypointlength):
        x=a[j]
        listx = x.tolist()
        x.sort()
        minval1=x[0]
        minval2=x[1]
        itemindex1 = listx.index(minval1)
        itemindex2 = listx.index(minval2)
```

```
        ratio = minval1/minval2

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

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

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

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

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

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

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

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

In [23]:

```
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[61:100]:
    right_files_path.append(folder_path + file)
```

In [24]:

```
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.INTE
R_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.INT
ER_CUBIC)
    images_right.append(cv2.cvtColor(right_img, cv2.COLOR_BGR2GRAY).astype('float32')/255
.)
    images_right_bgr.append(right_img)
```

```
100%|██████████| 61/61 [01:01<00:00,  1.00s/it]
100%|██████████| 39/39 [00:38<00:00,  1.00it/s]
```

In [25]:

```
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.INTE
R_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.INT
ER_CUBIC)
    images_right_bgr_no_enhance.append(right_img)
```

```
100%|██████████| 61/61 [00:25<00:00,  2.40it/s]
100%|██████████| 39/39 [00:17<00:00,  2.29it/s]
```

In [26]:

```
Threshl=60;
Octaves=8;
#PatternScales=1.0f;
brisk = cv2.BRISK_create(Threshl,Octaves)

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


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


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

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

```
100%|██████████| 61/61 [00:56<00:00,  1.09it/s]
100%|██████████| 39/39 [00:35<00:00,  1.11it/s]
```

In [27]:

```
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]))
```

```
100%|██████████| 61/61 [00:10<00:00,  5.99it/s]
100%|██████████| 39/39 [00:07<00:00,  5.44it/s]
```

In [28]:

```
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]))
```

```
100%|██████████| 61/61 [06:12<00:00,  6.11s/it]
100%|██████████| 39/39 [04:00<00:00,  6.17s/it]
```

In [29]:

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

In [30]:

```
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]))
```

```
100%|██████████| 61/61 [01:11<00:00,  1.16s/it]
100%|██████████| 39/39 [00:46<00:00,  1.18s/it]
```

In [48]:

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

```
Requirement already satisfied: opencv-python==3.4.2.17 in /opt/conda/lib/python3.7/site-p
ackages (3.4.2.17)
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-packages (f
rom opencv-python==3.4.2.17) (1.19.5)
Requirement already satisfied: opencv-contrib-python==3.4.2.17 in /opt/conda/lib/python3.
7/site-packages (3.4.2.17)
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-packages (f
rom opencv-contrib-python==3.4.2.17) (1.19.5)
```

In [51]:

```
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]))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-51-580f684962e0> in <module>
----> 1 star = cv2.xfeatures2d.StarDetector_create()
      2 brief = cv2.xfeatures2d.BriefDescriptorExtractor_create()
      3 keypoints_all_left_star = []
      4 descriptors_all_left_brief = []
      5 points_all_left_star=[]

AttributeError: module 'cv2.cv2' has no attribute 'xfeatures2d'
```

In [54]:

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


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

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

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

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-54-f20a3ddf3fc4> in <module>
      3 #PatternScales=1.0f;
      4 brisk = cv2.BRISK_create(Threshl,Octaves)
----> 5 freak = cv2.xfeatures2d.FREAK_create()
      6 keypoints_all_left_freak = []
      7 descriptors_all_left_freak = []

AttributeError: module 'cv2.cv2' has no attribute 'xfeatures2d'
```

In [55]:

```
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]))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-55-398c86fe01d3> in <module>
      1 mser = cv2.MSER_create()
----> 2 sift = cv2.xfeatures2d.SIFT_create()
      3 keypoints_all_left_mser = []
      4 descriptors_all_left_mser = []
      5 points_all_left_mser=[]

AttributeError: module 'cv2.cv2' has no attribute 'xfeatures2d'
```

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 [ ]:

```
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]))
```

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 [ ]:

```
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]))
    
```

In [88]:

```
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]))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-88-cb7ac11e2b7a> in <module>
----> 1 sift = cv2.xfeatures2d.SIFT_create()
      2 keypoints_all_left_sift = []
      3 descriptors_all_left_sift = []
      4 points_all_left_sift=[]
      5

AttributeError: module 'cv2.cv2' has no attribute 'xfeatures2d'
```

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 [89]:

```
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 [90]:

```
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]))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-90-fe73d8c8fcd1> in <module>
----> 1 sift = cv2.xfeatures2d.SIFT_create()
      2 rootsift = RootSIFT()
      3 keypoints_all_left_rootsift = []
      4 descriptors_all_left_rootsift = []
      5 points_all_left_rootsift=[]

AttributeError: module 'cv2.cv2' has no attribute 'xfeatures2d'
```

In [56]:

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

```
fatal: destination path 'SuperPointPretrainedNetwork' already exists and is not an empty
directory.
```

In [57]:

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

In [58]:

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

In [59]:

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


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

    def forward(self,x):
        x = self.relu(self.conv1a(x))
        x = self.relu(self.conv1b(x))
        x = self.pool(x)
        x = self.relu(self.conv2a(x))
        x = self.relu(self.conv2b(x))
        x = self.pool(x)
        x  = self.relu(self.conv3a(x))
```

```
        x = self.relu(self.conv3b(x))
        x = self.pool(x)
        x = self.relu(self.conv4a(x))
        x = self.relu(self.conv4b(x))
        cPa = self.relu(self.convPa(x))
        semi = self.convPb(cPa)
        cDa = self.relu(self.convDa(x))
        desc = self.convDb(cDa)
        dn = torch.norm(desc,p=2,dim=1)
        desc = desc.div(torch.unsqueeze(dn,1))
        return semi,desc


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

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

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


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

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

    def run(self,img):
        assert img.ndim == 2
        assert img.dtype == np.float32
        H,W = img.shape[0], img.shape[1]
```

```
        inp = img.copy()
        inp = (inp.reshape(1,H,W))
        inp = torch.from_numpy(inp)
        inp = torch.autograd.Variable(inp).view(1,1,H,W)
        if self.cuda:
            inp = inp.cuda()
        outs = self.net.forward(inp)
        semi,coarse_desc = outs[0],outs[1]
        semi = semi.data.cpu().numpy().squeeze()


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

        return heatmap,coarse_desc

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

        pts = np.zeros((3,len(xs)))
        pts[0,:] = ys
        pts[1,:] = xs
        pts[2,:] = heat_map[xs,ys]
        pts,_ = self.nms_fast(pts,H,W,dist_thresh=self.nms_dist)
        inds = np.argsort(pts[2,:])
        pts = pts[:,inds[::-1]]
        bord = self.border_remove
        toremoveW = np.logical_or(pts[0,:] < bord, pts[0,:] >= (W-bord))
        toremoveH = np.logical_or(pts[1,:] < bord, pts[0,:] >= (H-bord))
        toremove = np.logical_or(toremoveW, toremoveH)
        pts = pts[:,~toremove]
        pts = pts[:,0:sampled]
        D = coarse_desc.shape[1]
        if pts.shape[1] == 0:
            desc = np.zeros((D,0))
        else:
            samp_pts = torch.from_numpy(pts[:2,:].copy())
            samp_pts[0,:] = (samp_pts[0,:] / (float(W)/2.))-1.
            samp_pts[1,:] = (samp_pts[1,:] / (float(W)/2.))-1.
            samp_pts = samp_pts.transpose(0,1).contiguous()
            samp_pts = samp_pts.view(1,1,-1,2)
            samp_pts = samp_pts.float()
            if self.cuda:
                samp_pts = samp_pts.cuda()
            desc = nn.functional.grid_sample(coarse_desc, samp_pts)
            desc = desc.data.cpu().numpy().reshape(D,-1)
            desc /= np.linalg.norm(desc,axis=0)[np.newaxis,:]
        return pts,desc
```

In [60]:

```
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 [71]:

```
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)
```

```
  0%|          | 0/61 [00:00<?, ?it/s]
Number of pts selected: 37731
  2%|▏         | 1/61 [00:00<00:26,  2.30it/s]
Number of pts selected: 39992
  3%|▎         | 2/61 [00:00<00:24,  2.43it/s]
Number of pts selected: 43330
  5%|▍         | 3/61 [00:01<00:23,  2.43it/s]
Number of pts selected: 44808
  7%|▋         | 4/61 [00:01<00:23,  2.41it/s]
Number of pts selected: 42649
  8%|▊         | 5/61 [00:02<00:23,  2.43it/s]
Number of pts selected: 41303
 10%|▉         | 6/61 [00:02<00:22,  2.43it/s]
Number of pts selected: 43513
 11%|█         | 7/61 [00:02<00:22,  2.41it/s]
Number of pts selected: 50045
 13%|█▏        | 8/61 [00:03<00:22,  2.35it/s]
Number of pts selected: 43866
 15%|█▍        | 9/61 [00:03<00:22,  2.36it/s]
Number of pts selected: 56533
 16%|█▋        | 10/61 [00:04<00:22,  2.25it/s]
Number of pts selected: 54470
 18%|█▊        | 11/61 [00:04<00:22,  2.20it/s]
Number of pts selected: 64536
```

Number of pts selected: 61990
 20%|██ | 12/61 [00:05<00:23, 2.08it/s]
Number of pts selected: 62738
 21%|██ | 13/61 [00:05<00:23, 2.02it/s]
Number of pts selected: 66925
 23%|██ | 14/61 [00:06<00:26, 1.74it/s]
Number of pts selected: 65946
 25%|██ | 15/61 [00:07<00:26, 1.77it/s]
Number of pts selected: 62358
 26%|██ | 16/61 [00:07<00:24, 1.80it/s]
Number of pts selected: 58450
 28%|██ | 17/61 [00:08<00:23, 1.84it/s]
Number of pts selected: 52783
 30%|██ | 18/61 [00:08<00:22, 1.92it/s]
Number of pts selected: 53712
 31%|███ | 19/61 [00:09<00:21, 1.96it/s]
Number of pts selected: 52893
 33%|███ | 20/61 [00:09<00:20, 2.01it/s]
Number of pts selected: 48246
 34%|███ | 21/61 [00:10<00:19, 2.07it/s]
Number of pts selected: 53586
 36%|███ | 22/61 [00:10<00:18, 2.08it/s]
Number of pts selected: 53274
 38%|███ | 23/61 [00:10<00:18, 2.08it/s]
Number of pts selected: 55579
 39%|███ | 24/61 [00:11<00:17, 2.07it/s]
Number of pts selected: 53750
 41%|████ | 25/61 [00:11<00:17, 2.08it/s]
Number of pts selected: 63693
 43%|████ | 26/61 [00:12<00:18, 1.85it/s]
Number of pts selected: 63825
 44%|████ | 27/61 [00:13<00:20, 1.63it/s]
Number of pts selected: 63398
 46%|████ | 28/61 [00:13<00:19, 1.70it/s]
Number of pts selected: 57589
 48%|████ | 29/61 [00:14<00:18, 1.78it/s]
Number of pts selected: 45296
 49%|████ | 30/61 [00:14<00:16, 1.90it/s]
Number of pts selected: 43178
 51%|████ | 31/61 [00:15<00:14, 2.03it/s]
Number of pts selected: 32932

Number of pts selected: 52932

52%|[redacted]| 32/61 [00:15<00:13, 2.22it/s]

Number of pts selected: 36236

54%|[redacted]| 33/61 [00:16<00:11, 2.34it/s]

Number of pts selected: 36092

56%|[redacted]| 34/61 [00:16<00:11, 2.44it/s]

Number of pts selected: 35489

57%|[redacted]| 35/61 [00:16<00:10, 2.52it/s]

Number of pts selected: 41577

59%|[redacted]| 36/61 [00:17<00:10, 2.35it/s]

Number of pts selected: 41828

61%|[redacted]| 37/61 [00:17<00:10, 2.28it/s]

Number of pts selected: 52538

62%|[redacted]| 38/61 [00:18<00:10, 2.20it/s]

Number of pts selected: 51649

64%|[redacted]| 39/61 [00:18<00:10, 2.19it/s]

Number of pts selected: 53866

66%|[redacted]| 40/61 [00:19<00:09, 2.16it/s]

Number of pts selected: 51578

67%|[redacted]| 41/61 [00:19<00:09, 2.15it/s]

Number of pts selected: 52527

69%|[redacted]| 42/61 [00:20<00:08, 2.15it/s]

Number of pts selected: 53350

70%|[redacted]| 43/61 [00:20<00:08, 2.15it/s]

Number of pts selected: 58951

72%|[redacted]| 44/61 [00:21<00:08, 2.10it/s]

Number of pts selected: 63675

74%|[redacted]| 45/61 [00:21<00:07, 2.04it/s]

Number of pts selected: 63148

75%|[redacted]| 46/61 [00:22<00:07, 2.00it/s]

Number of pts selected: 61328

77%|[redacted]| 47/61 [00:22<00:07, 1.97it/s]

Number of pts selected: 62622

79%|[redacted]| 48/61 [00:23<00:06, 1.96it/s]

Number of pts selected: 60923

80%|[redacted]| 49/61 [00:23<00:06, 1.95it/s]

Number of pts selected: 59770

82%|[redacted]| 50/61 [00:24<00:05, 1.96it/s]

Number of pts selected: 59411

84%|[redacted]| 51/61 [00:24<00:05, 1.95it/s]

Number of pts selected: 57028

Number of pts selected: 57020

 85%|████████ | 52/61 [00:25<00:04, 1.97it/s]

Number of pts selected: 60032

 87%|████████▏ | 53/61 [00:25<00:04, 1.96it/s]

Number of pts selected: 60506

 89%|████████▏ | 54/61 [00:26<00:03, 1.96it/s]

Number of pts selected: 61336

 90%|████████▉ | 55/61 [00:26<00:03, 1.95it/s]

Number of pts selected: 60851

 92%|█████████ | 56/61 [00:27<00:02, 1.94it/s]

Number of pts selected: 57779

 93%|█████████▏| 57/61 [00:27<00:02, 1.95it/s]

Number of pts selected: 59678

 95%|█████████▍| 58/61 [00:28<00:01, 1.80it/s]

Number of pts selected: 58116

 97%|█████████▌| 59/61 [00:28<00:01, 1.84it/s]

Number of pts selected: 59898

 98%|█████████▊| 60/61 [00:29<00:00, 1.87it/s]

Number of pts selected: 57343

100%|██████████| 61/61 [00:29<00:00, 2.04it/s]
  0%| | 0/39 [00:00<?, ?it/s]

Number of pts selected: 41060

  3%|▎ | 1/39 [00:00<00:15, 2.51it/s]

Number of pts selected: 45916

  5%|▌ | 2/39 [00:00<00:15, 2.41it/s]

Number of pts selected: 49163

  8%|▊ | 3/39 [00:01<00:15, 2.34it/s]

Number of pts selected: 57449

 10%|█ | 4/39 [00:01<00:15, 2.21it/s]

Number of pts selected: 55449

 13%|█▎ | 5/39 [00:02<00:15, 2.15it/s]

Number of pts selected: 52593

 15%|█▌ | 6/39 [00:02<00:15, 2.15it/s]

Number of pts selected: 55416

 18%|█▊ | 7/39 [00:03<00:15, 2.12it/s]

Number of pts selected: 54325

 21%|██ | 8/39 [00:03<00:14, 2.08it/s]

Number of pts selected: 51642

 23%|██▎ | 9/39 [00:04<00:14, 2.10it/s]

Number of pts selected: 50175

 26%|██▌ | 10/39 [00:04<00:13, 2.14it/s]

```
Number of pts selected: 48680
 28%|██▉       | 11/39 [00:05<00:12,  2.17it/s]
Number of pts selected: 46066
 31%|███       | 12/39 [00:05<00:12,  2.22it/s]
Number of pts selected: 48052
 33%|███▎      | 13/39 [00:05<00:11,  2.24it/s]
Number of pts selected: 57649
 36%|███▌      | 14/39 [00:06<00:11,  2.16it/s]
Number of pts selected: 70769
 38%|███▊      | 15/39 [00:06<00:11,  2.01it/s]
Number of pts selected: 72475
 41%|████      | 16/39 [00:07<00:12,  1.91it/s]
Number of pts selected: 75193
 44%|████▎     | 17/39 [00:08<00:12,  1.82it/s]
Number of pts selected: 72208
 46%|████▌     | 18/39 [00:08<00:11,  1.78it/s]
Number of pts selected: 66153
 49%|████▊     | 19/39 [00:09<00:12,  1.65it/s]
Number of pts selected: 59368
 51%|█████     | 20/39 [00:09<00:10,  1.73it/s]
Number of pts selected: 61077
 54%|█████▎    | 21/39 [00:10<00:10,  1.79it/s]
Number of pts selected: 50082
 56%|█████▌    | 22/39 [00:10<00:09,  1.88it/s]
Number of pts selected: 33880
 59%|█████▊    | 23/39 [00:11<00:07,  2.06it/s]
Number of pts selected: 32749
 62%|██████    | 24/39 [00:11<00:06,  2.25it/s]
Number of pts selected: 36259
 64%|██████▎   | 25/39 [00:12<00:05,  2.36it/s]
Number of pts selected: 50132
 67%|██████▌   | 26/39 [00:12<00:05,  2.31it/s]
Number of pts selected: 47551
 69%|██████▊   | 27/39 [00:12<00:05,  2.30it/s]
Number of pts selected: 54679
 72%|███████   | 28/39 [00:13<00:04,  2.24it/s]
Number of pts selected: 60352
 74%|███████▎  | 29/39 [00:13<00:04,  2.14it/s]
Number of pts selected: 73857
 77%|███████▌  | 30/39 [00:14<00:04,  1.99it/s]
```

```
Number of pts selected: 72114
 79%|███████▉  | 31/39 [00:15<00:04,  1.65it/s]
Number of pts selected: 70208
 82%|████████  | 32/39 [00:16<00:04,  1.60it/s]
Number of pts selected: 65683
 85%|████████▍ | 33/39 [00:16<00:03,  1.66it/s]
Number of pts selected: 63707
 87%|████████▋ | 34/39 [00:17<00:02,  1.71it/s]
Number of pts selected: 57070
 90%|████████▉ | 35/39 [00:17<00:02,  1.79it/s]
Number of pts selected: 44606
 92%|█████████▏| 36/39 [00:18<00:01,  1.93it/s]
Number of pts selected: 42460
 95%|█████████▍| 37/39 [00:18<00:00,  2.06it/s]
Number of pts selected: 41800
 97%|█████████▋| 38/39 [00:18<00:00,  2.17it/s]
Number of pts selected: 43585
100%|██████████| 39/39 [00:19<00:00,  2.02it/s]
```

In [72]:

```
num_kps_brisk  = []
num_kps_orb = []
num_kps_akaze  = []
num_kps_kaze = []

for j in tqdm(keypoints_all_left_brisk + keypoints_all_right_brisk):
    num_kps_surf.append(len(j))

for j  in tqdm(keypoints_all_left_orb + keypoints_all_right_orb):
    num_kps_orb.append(len(j))

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

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

```
100%|██████████| 100/100 [00:00<00:00, 419430.40it/s]
100%|██████████| 100/100 [00:00<00:00, 375833.69it/s]
100%|██████████| 100/100 [00:00<00:00, 477711.16it/s]
100%|██████████| 100/100 [00:00<00:00, 416514.80it/s]
```

In [82]:

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

In [86]:

```
def get_Hmatrix(imgs,keypts,pts,descripts,ratio=0.8,thresh=4,disp=False):
```

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

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

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

    inlier_matchset = np.array(matches_4)[inliers.astype(bool)].tolist()
    print("Number of Robust matches",len(inlier_matchset))
    print("\n")
    '''
    if len(inlier_matchset)<50:
        matches_4 = []
        ratio = 0.67
        # loop over the raw matches
        for m in matches_lf1_lf:
           # ensure the distance is within a certain ratio of each
           # other (i.e. Lowe's ratio test)
           if len(m) == 2 and m[0].distance < m[1].distance * ratio:
           #matches_1.append((m[0].trainIdx, m[0].queryIdx))
           matches_4.append(m[0])
        print("Number of matches After Lowe's Ratio New",len(matches_4))
        matches_idx = np.array([m.queryIdx for m in matches_4])
        imm1_pts = np.array([keypts[0][idx].pt for idx in matches_idx])
        matches_idx = np.array([m.trainIdx for m in matches_4])
        imm2_pts = np.array([keypts[1][idx].pt for idx in matches_idx])
        Hn,inliers = compute_homography_fast_other(imm1_pts,imm2_pts)
        inlier_matchset = np.array(matches_4)[inliers.astype(bool)].tolist()
        print("Number of Robust matches New",len(inlier_matchset))
        print("\n")
       '''
    #H=compute_Homography(imm1_pts,imm2_pts)
    #Robustly estimate Homography 1 using RANSAC
    #Hn=RANSAC_alg(keypts[0] ,keypts[1], matches_4, nRANSAC=1500, RANSACthresh=6)
    #global inlier_matchset
```

```
    if disp==True:
        dispimg1=cv2.drawMatches(imgs[0], keypts[0], imgs[1], keypts[1], inlier_matchset
, None,flags=2)
        displayplot(dispimg1,'Robust Matching between Reference Image and Right Image ')
    return Hn/Hn[2,2], len(matches_lf1_lf), len(inlier_matchset)
```

In [84]:

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

In [77]:

```
len(images_left)
```

Out[77]:

61

In [91]:

```
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 [00:05<05:58,  5.97s/it]
```

```
Number of matches 25071
Number of matches After Lowe's Ratio 841
Number of Robust matches 341
```

```
  3%|          | 2/61 [00:12<06:14,  6.35s/it]
```

```
Number of matches 30932
Number of matches After Lowe's Ratio 733
Number of Robust matches 248
```

```
  5%|          | 3/61 [00:21<07:07,  7.37s/it]
```

```
Number of matches 26045
Number of matches After Lowe's Ratio 295
Number of Robust matches 36
```

```
  7%|▋         | 4/61 [00:27<06:40,  7.03s/it]
Number of matches 23435
Number of matches After Lowe's Ratio 1603
Number of Robust matches 906


  8%|▊         | 5/61 [00:33<06:18,  6.76s/it]
Number of matches 28290
Number of matches After Lowe's Ratio 2013
Number of Robust matches 1089


 10%|▉         | 6/61 [00:41<06:23,  6.98s/it]
Number of matches 26533
Number of matches After Lowe's Ratio 1742
Number of Robust matches 937


 11%|█         | 7/61 [00:48<06:22,  7.08s/it]
Number of matches 32308
Number of matches After Lowe's Ratio 2176
Number of Robust matches 1333


 13%|█▏        | 8/61 [00:57<06:44,  7.64s/it]
Number of matches 22852
Number of matches After Lowe's Ratio 1065
Number of Robust matches 572


 15%|█▎        | 9/61 [01:03<06:10,  7.12s/it]
Number of matches 31117
Number of matches After Lowe's Ratio 1790
Number of Robust matches 1178


 16%|█▍        | 10/61 [01:11<06:17,  7.40s/it]
Number of matches 26415
Number of matches After Lowe's Ratio 1214
Number of Robust matches 802


 18%|█▌        | 11/61 [01:18<06:04,  7.29s/it]
Number of matches 32660
Number of matches After Lowe's Ratio 2203
Number of Robust matches 1562


 20%|█▋        | 12/61 [01:28<06:29,  7.95s/it]
Number of matches 32145
Number of matches After Lowe's Ratio 2587
Number of Robust matches 1890


 21%|█▊        | 13/61 [01:37<06:37,  8.27s/it]
Number of matches 37729
```

```
Number of matches After Lowe's Ratio 2591
Number of Robust matches 1855


 23%|██        | 14/61 [01:47<07:06,  9.07s/it]

Number of matches 37557
Number of matches After Lowe's Ratio 3692
Number of Robust matches 2900


 25%|██        | 15/61 [01:58<07:23,  9.63s/it]

Number of matches 34855
Number of matches After Lowe's Ratio 2804
Number of Robust matches 2144


 26%|██        | 16/61 [02:08<07:10,  9.57s/it]

Number of matches 30877
Number of matches After Lowe's Ratio 2833
Number of Robust matches 2184


 28%|██        | 17/61 [02:16<06:47,  9.26s/it]

Number of matches 32047
Number of matches After Lowe's Ratio 2662
Number of Robust matches 2031


 30%|███       | 18/61 [02:25<06:36,  9.22s/it]

Number of matches 33091
Number of matches After Lowe's Ratio 3145
Number of Robust matches 2211


Number of matches 31279
Number of matches After Lowe's Ratio 3367
 31%|███       | 19/61 [02:35<06:28,  9.26s/it]

Number of Robust matches 2534


 33%|███       | 20/61 [02:43<06:11,  9.05s/it]

Number of matches 31422
Number of matches After Lowe's Ratio 2503
Number of Robust matches 1737


 34%|███       | 21/61 [02:52<05:52,  8.82s/it]

Number of matches 32143
Number of matches After Lowe's Ratio 2050
Number of Robust matches 1323


 36%|███       | 22/61 [03:01<05:49,  8.96s/it]

Number of matches 31819
Number of matches After Lowe's Ratio 2185
Number of Robust matches 1380
```

38%|███▊ | 23/61 [03:10<05:38, 8.90s/it]

```
Number of matches 33063
Number of matches After Lowe's Ratio 2379
Number of Robust matches 1641
```

39%|███▉ | 24/61 [03:19<05:32, 8.98s/it]

```
Number of matches 35871
Number of matches After Lowe's Ratio 1955
Number of Robust matches 1337
```

41%|████ | 25/61 [03:30<05:42, 9.52s/it]

```
Number of matches 43925
Number of matches After Lowe's Ratio 2222
Number of Robust matches 1037
```

43%|████▎ | 26/61 [03:42<06:05, 10.44s/it]

```
Number of matches 37982
Number of matches After Lowe's Ratio 1848
Number of Robust matches 812
```

44%|████▍ | 27/61 [03:53<05:57, 10.53s/it]

```
Number of matches 32060
Number of matches After Lowe's Ratio 1846
Number of Robust matches 1039
```

46%|████▌ | 28/61 [04:02<05:30, 10.02s/it]

```
Number of matches 30283
Number of matches After Lowe's Ratio 1569
Number of Robust matches 614
```

48%|████▊ | 29/61 [04:10<05:04, 9.51s/it]

```
Number of matches 33737
Number of matches After Lowe's Ratio 995
Number of Robust matches 420
```

49%|████▉ | 30/61 [04:20<04:54, 9.50s/it]

```
Number of matches 34231
Number of matches After Lowe's Ratio 1419
Number of Robust matches 629
```

51%|█████ | 31/61 [04:30<04:52, 9.77s/it]

```
Number of matches 34536
Number of matches After Lowe's Ratio 661
Number of Robust matches 239
```

52%|█████▏ | 32/61 [04:40<04:43, 9.79s/it]

```
Number of matches 24595
Number of matches After Lowe's Ratio 343
Number of Robust matches 48


 54%|[redacted]     | 33/61 [04:46<04:05,  8.78s/it]

Number of matches 24705
Number of matches After Lowe's Ratio 1382
Number of Robust matches 732


 56%|[redacted]     | 34/61 [04:53<03:37,  8.07s/it]

Number of matches 21203
Number of matches After Lowe's Ratio 1399
Number of Robust matches 708


 57%|[redacted]     | 35/61 [04:58<03:10,  7.33s/it]

Number of matches 26268
Number of matches After Lowe's Ratio 1374
Number of Robust matches 732


 59%|[redacted]     | 36/61 [05:06<03:03,  7.34s/it]

Number of matches 31991
Number of matches After Lowe's Ratio 1781
Number of Robust matches 797


 61%|[redacted]     | 37/61 [05:15<03:11,  7.98s/it]

Number of matches 44712
Number of matches After Lowe's Ratio 1749
Number of Robust matches 704


 62%|[redacted]     | 38/61 [05:29<03:43,  9.72s/it]

Number of matches 48503
Number of matches After Lowe's Ratio 2349
Number of Robust matches 699


 64%|[redacted]     | 39/61 [05:44<04:12, 11.47s/it]

Number of matches 44596
Number of matches After Lowe's Ratio 2159
Number of Robust matches 749


 66%|[redacted]     | 40/61 [05:58<04:12, 12.05s/it]

Number of matches 35513
Number of matches After Lowe's Ratio 2148
Number of Robust matches 1080


 67%|[redacted]     | 41/61 [06:08<03:48, 11.40s/it]

Number of matches 32918
Number of matches After Lowe's Ratio 2626
Number of Robust matches 1512
```

69%|██████▉ | 42/61 [06:18<03:28, 10.96s/it]

Number of matches 30919
Number of matches After Lowe's Ratio 2645
Number of Robust matches 1870

70%|███████ | 43/61 [06:26<03:04, 10.25s/it]

Number of matches 30527
Number of matches After Lowe's Ratio 2867
Number of Robust matches 2002

Number of matches 36841
Number of matches After Lowe's Ratio 2662

72%|███████▏ | 44/61 [06:35<02:46, 9.78s/it]

Number of Robust matches 1592

74%|███████▍ | 45/61 [06:46<02:42, 10.17s/it]

Number of matches 39752
Number of matches After Lowe's Ratio 3145
Number of Robust matches 1848

75%|███████▌ | 46/61 [06:58<02:39, 10.64s/it]

Number of matches 38017
Number of matches After Lowe's Ratio 3197
Number of Robust matches 1853

77%|███████▋ | 47/61 [07:09<02:31, 10.80s/it]

Number of matches 39813
Number of matches After Lowe's Ratio 3052
Number of Robust matches 1715

79%|███████▊ | 48/61 [07:21<02:23, 11.07s/it]

Number of matches 33280
Number of matches After Lowe's Ratio 2128
Number of Robust matches 1327

80%|████████ | 49/61 [07:30<02:05, 10.49s/it]

Number of matches 31740
Number of matches After Lowe's Ratio 3613
Number of Robust matches 2570

82%|████████▏ | 50/61 [07:39<01:50, 10.02s/it]

Number of matches 31022
Number of matches After Lowe's Ratio 3233
Number of Robust matches 2167

84%|▉ | 51/61 [07:47<01:35, 9.54s/it]

```
Number of matches 27874
Number of matches After Lowe's Ratio 1803
Number of Robust matches 1013
```

85%|▉ | 52/61 [07:54<01:19, 8.88s/it]

```
Number of matches 27354
Number of matches After Lowe's Ratio 1799
Number of Robust matches 1261
```

87%|▉ | 53/61 [08:02<01:07, 8.46s/it]

```
Number of matches 27672
Number of matches After Lowe's Ratio 2483
Number of Robust matches 1865
```

89%|▉ | 54/61 [08:09<00:56, 8.11s/it]

```
Number of matches 32783
Number of matches After Lowe's Ratio 2191
Number of Robust matches 1249
```

90%|▉ | 55/61 [08:18<00:50, 8.38s/it]

```
Number of matches 27058
Number of matches After Lowe's Ratio 2165
Number of Robust matches 1296
```

92%|▉ | 56/61 [08:26<00:40, 8.12s/it]

```
Number of matches 29170
Number of matches After Lowe's Ratio 1893
Number of Robust matches 950
```

93%|▉ | 57/61 [08:34<00:32, 8.06s/it]

```
Number of matches 32843
Number of matches After Lowe's Ratio 2898
Number of Robust matches 1493
```

95%|▉ | 58/61 [08:43<00:24, 8.30s/it]

```
Number of matches 33022
Number of matches After Lowe's Ratio 1907
Number of Robust matches 717
```

97%|▉ | 59/61 [08:52<00:17, 8.78s/it]

```
Number of matches 35941
Number of matches After Lowe's Ratio 2903
Number of Robust matches 1048
```

98%|▉ | 60/61 [09:02<00:09, 9.05s/it]
0%| | 0/39 [00:00<?, ?it/s]

```
Number of matches 25328
```

```
Number of matches After Lowe's Ratio 805
Number of Robust matches 259


  3%|▍          | 1/39 [00:06<04:12,  6.65s/it]

Number of matches 35142
Number of matches After Lowe's Ratio 1885
Number of Robust matches 1361


  5%|▌          | 2/39 [00:16<05:15,  8.53s/it]

Number of matches 28733
Number of matches After Lowe's Ratio 2290
Number of Robust matches 1766


  8%|▊          | 3/39 [00:24<04:53,  8.16s/it]

Number of matches 25875
Number of matches After Lowe's Ratio 1261
Number of Robust matches 835


 10%|▉          | 4/39 [00:30<04:25,  7.58s/it]

Number of matches 24358
Number of matches After Lowe's Ratio 560
Number of Robust matches 276


 13%|█▏         | 5/39 [00:36<03:57,  6.99s/it]

Number of matches 21252
Number of matches After Lowe's Ratio 1666
Number of Robust matches 1266


 15%|█▍         | 6/39 [00:42<03:35,  6.53s/it]

Number of matches 31075
Number of matches After Lowe's Ratio 1090
Number of Robust matches 651


 18%|█▋         | 7/39 [00:50<03:49,  7.17s/it]

Number of matches 31868
Number of matches After Lowe's Ratio 2904
Number of Robust matches 2363


 21%|█▉         | 8/39 [01:00<04:01,  7.80s/it]

Number of matches 33234
Number of matches After Lowe's Ratio 3147
Number of Robust matches 2611


 23%|██▏        | 9/39 [01:09<04:05,  8.18s/it]

Number of matches 29690
Number of matches After Lowe's Ratio 2538
Number of Robust matches 2038
```

```
 26%|██        | 10/39 [01:17<03:57,  8.18s/it]
Number of matches 32999
Number of matches After Lowe's Ratio 2609
Number of Robust matches 2052

 28%|██▊       | 11/39 [01:26<03:59,  8.56s/it]
Number of matches 33074
Number of matches After Lowe's Ratio 1890
Number of Robust matches 1362

 31%|███       | 12/39 [01:36<03:57,  8.81s/it]
Number of matches 35105
Number of matches After Lowe's Ratio 2325
Number of Robust matches 1557

 33%|███▎      | 13/39 [01:46<03:58,  9.17s/it]
Number of matches 39186
Number of matches After Lowe's Ratio 2340
Number of Robust matches 1580

 36%|███▌      | 14/39 [01:57<04:08,  9.92s/it]
Number of matches 37695
Number of matches After Lowe's Ratio 2631
Number of Robust matches 1445

 38%|███▊      | 15/39 [02:08<04:04, 10.19s/it]
Number of matches 40599
Number of matches After Lowe's Ratio 2823
Number of Robust matches 1454

 41%|████      | 16/39 [02:20<04:05, 10.66s/it]
Number of matches 35734
Number of matches After Lowe's Ratio 2659
Number of Robust matches 1408

 44%|████▎     | 17/39 [02:30<03:51, 10.54s/it]
Number of matches 29132
Number of matches After Lowe's Ratio 2051
Number of Robust matches 926

 46%|████▌     | 18/39 [02:38<03:23,  9.71s/it]
Number of matches 31566
Number of matches After Lowe's Ratio 2662
Number of Robust matches 1052

 49%|████▊     | 19/39 [02:47<03:07,  9.38s/it]
Number of matches 30048
```

Number of matches 30040
Number of matches After Lowe's Ratio 2205
Number of Robust matches 781

51%|[redacted]| 20/39 [02:54<02:50, 8.95s/it]

Number of matches 24007
Number of matches After Lowe's Ratio 1625
Number of Robust matches 666

54%|[redacted]| 21/39 [03:01<02:27, 8.22s/it]

Number of matches 29055
Number of matches After Lowe's Ratio 1447
Number of Robust matches 785

56%|[redacted]| 22/39 [03:09<02:19, 8.23s/it]

Number of matches 45487
Number of matches After Lowe's Ratio 564
Number of Robust matches 130

59%|[redacted]| 23/39 [03:23<02:37, 9.82s/it]

Number of matches 41464
Number of matches After Lowe's Ratio 1108
Number of Robust matches 443

62%|[redacted]| 24/39 [03:36<02:40, 10.72s/it]

Number of matches 47398
Number of matches After Lowe's Ratio 309
Number of Robust matches 6

64%|[redacted]| 25/39 [03:50<02:43, 11.69s/it]

Number of matches 36716
Number of matches After Lowe's Ratio 1036
Number of Robust matches 360

67%|[redacted]| 26/39 [04:00<02:26, 11.29s/it]

Number of matches 34503
Number of matches After Lowe's Ratio 1960
Number of Robust matches 847

69%|[redacted]| 27/39 [04:10<02:11, 10.93s/it]

Number of matches 32306
Number of matches After Lowe's Ratio 2087
Number of Robust matches 765

72%|[redacted]| 28/39 [04:19<01:52, 10.25s/it]

Number of matches 28440
Number of matches After Lowe's Ratio 1658
Number of Robust matches 637

74%|███████▍ | 29/39 [04:26<01:33, 9.35s/it]

```
Number of matches 26060
Number of matches After Lowe's Ratio 1373
Number of Robust matches 496
```

77%|███████▋ | 30/39 [04:33<01:17, 8.60s/it]

```
Number of matches 27600
Number of matches After Lowe's Ratio 1348
Number of Robust matches 435
```

79%|███████▉ | 31/39 [04:40<01:06, 8.36s/it]

```
Number of matches 29911
Number of matches After Lowe's Ratio 2472
Number of Robust matches 809
```

82%|████████▏ | 32/39 [04:48<00:57, 8.24s/it]

```
Number of matches 31772
Number of matches After Lowe's Ratio 1394
Number of Robust matches 481
```

85%|████████▍ | 33/39 [04:57<00:50, 8.34s/it]

```
Number of matches 28612
Number of matches After Lowe's Ratio 2171
Number of Robust matches 1017
```

87%|████████▋ | 34/39 [05:05<00:41, 8.26s/it]

```
Number of matches 32692
Number of matches After Lowe's Ratio 1979
Number of Robust matches 743
```

90%|████████▉ | 35/39 [05:14<00:34, 8.56s/it]

```
Number of matches 25607
Number of matches After Lowe's Ratio 1495
Number of Robust matches 725
```

92%|█████████▏ | 36/39 [05:21<00:23, 7.96s/it]

```
Number of matches 28478
Number of matches After Lowe's Ratio 1131
Number of Robust matches 770
```

95%|█████████▍ | 37/39 [05:29<00:15, 7.89s/it]

```
Number of matches 28772
Number of matches After Lowe's Ratio 1473
Number of Robust matches 908
```

97%|█████████▋ | 38/39 [05:36<00:08, 8.86s/it]

```
Number of matches 27878
Number of matches After Lowe's Ratio 1479
Number of Robust matches 950
```

In [93]:

```
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
_brisk[j:j+2][::-1],points_all_left_brisk[j:j+2][::-1],descriptors_all_left_brisk[j:j+2]
[::-1])
    H_left_orb.append(H_a)
    num_matches_orb.append(matches)
    num_good_matches_orb.append(gd_matches)

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

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

```
  2%|          | 1/61 [00:05<05:51,  5.86s/it]
Number of matches 25071
Number of matches After Lowe's Ratio 856
Number of Robust matches 368

  3%|          | 2/61 [00:12<06:13,  6.33s/it]
Number of matches 30932
Number of matches After Lowe's Ratio 712
Number of Robust matches 271

  5%|          | 3/61 [00:21<07:10,  7.43s/it]
Number of matches 26045
Number of matches After Lowe's Ratio 318
Number of Robust matches 33

  7%|          | 4/61 [00:27<06:47,  7.15s/it]
Number of matches 23435
Number of matches After Lowe's Ratio 1574
Number of Robust matches 860

  8%|          | 5/61 [00:33<06:15,  6.70s/it]
Number of matches 28290
Number of matches After Lowe's Ratio 2034
Number of Robust matches 1073
```

```
 10%|          | 6/61 [00:41<06:28,  7.06s/it]
Number of matches 26533
Number of matches After Lowe's Ratio 1700
Number of Robust matches 830

 11%|          | 7/61 [00:48<06:26,  7.15s/it]
Number of matches 32308
Number of matches After Lowe's Ratio 2172
Number of Robust matches 1030

 13%|          | 8/61 [00:57<06:43,  7.61s/it]
Number of matches 22852
Number of matches After Lowe's Ratio 1066
Number of Robust matches 506

 15%|          | 9/61 [01:03<06:12,  7.17s/it]
Number of matches 31117
Number of matches After Lowe's Ratio 1761
Number of Robust matches 1133

 16%|          | 10/61 [01:11<06:22,  7.50s/it]
Number of matches 26415
Number of matches After Lowe's Ratio 1220
Number of Robust matches 691

 18%|          | 11/61 [01:18<06:05,  7.31s/it]
Number of matches 32660
Number of matches After Lowe's Ratio 2222
Number of Robust matches 1489

 20%|          | 12/61 [01:28<06:32,  8.00s/it]
Number of matches 32145
Number of matches After Lowe's Ratio 2560
Number of Robust matches 1904

 21%|          | 13/61 [01:37<06:37,  8.29s/it]
Number of matches 37729
Number of matches After Lowe's Ratio 2560
Number of Robust matches 1878

 23%|          | 14/61 [01:48<07:05,  9.06s/it]
Number of matches 37557
Number of matches After Lowe's Ratio 3695
Number of Robust matches 2884

 25%|          | 15/61 [01:59<07:26,  9.71s/it]
```

```
Number of matches 34855
Number of matches After Lowe's Ratio 2803
Number of Robust matches 2033


 26%|██        | 16/61 [02:09<07:14,  9.66s/it]

Number of matches 30877
Number of matches After Lowe's Ratio 2848
Number of Robust matches 2133


 28%|██        | 17/61 [02:17<06:46,  9.23s/it]

Number of matches 32047
Number of matches After Lowe's Ratio 2665
Number of Robust matches 1938


 30%|██        | 18/61 [02:26<06:38,  9.28s/it]

Number of matches 33091
Number of matches After Lowe's Ratio 3157
Number of Robust matches 2539


 31%|███       | 19/61 [02:35<06:27,  9.22s/it]

Number of matches 31279
Number of matches After Lowe's Ratio 3310
Number of Robust matches 2618


 33%|███       | 20/61 [02:44<06:12,  9.08s/it]

Number of matches 31422
Number of matches After Lowe's Ratio 2550
Number of Robust matches 1545


 34%|███       | 21/61 [02:53<05:56,  8.92s/it]

Number of matches 32143
Number of matches After Lowe's Ratio 2007
Number of Robust matches 1384


 36%|███       | 22/61 [03:02<05:52,  9.03s/it]

Number of matches 31819
Number of matches After Lowe's Ratio 2235
Number of Robust matches 1364


 38%|███       | 23/61 [03:11<05:39,  8.94s/it]

Number of matches 33063
Number of matches After Lowe's Ratio 2368
Number of Robust matches 1516


 39%|███       | 24/61 [03:20<05:34,  9.03s/it]

Number of matches 35871
Number of matches After Lowe's Ratio 1984
Number of Robust matches 1364
```

41%|████ | 25/61 [03:31<05:44, 9.57s/it]

```
Number of matches 43925
Number of matches After Lowe's Ratio 2307
Number of Robust matches 1047
```

43%|████ | 26/61 [03:43<06:09, 10.55s/it]

```
Number of matches 37982
Number of matches After Lowe's Ratio 1824
Number of Robust matches 824
```

44%|████ | 27/61 [03:54<05:58, 10.55s/it]

```
Number of matches 32060
Number of matches After Lowe's Ratio 1829
Number of Robust matches 877
```

46%|████ | 28/61 [04:03<05:34, 10.13s/it]

```
Number of matches 30283
Number of matches After Lowe's Ratio 1560
Number of Robust matches 628
```

48%|████ | 29/61 [04:12<05:08, 9.64s/it]

```
Number of matches 33737
Number of matches After Lowe's Ratio 948
Number of Robust matches 402
```

49%|████ | 30/61 [04:21<04:56, 9.56s/it]

```
Number of matches 34231
Number of matches After Lowe's Ratio 1384
Number of Robust matches 607
```

51%|█████ | 31/61 [04:31<04:47, 9.59s/it]

```
Number of matches 34536
Number of matches After Lowe's Ratio 658
Number of Robust matches 219
```

52%|█████ | 32/61 [04:40<04:39, 9.64s/it]

```
Number of matches 24595
Number of matches After Lowe's Ratio 342
Number of Robust matches 39
```

54%|█████ | 33/61 [04:47<04:00, 8.58s/it]

```
Number of matches 24705
Number of matches After Lowe's Ratio 1414
Number of Robust matches 694
```

56%|█████ | 34/61 [04:53<03:33, 7.90s/it]

```
Number of matches 21203
Number of matches After Lowe's Ratio 1360
Number of Robust matches 771


 57%|[redacted]     | 35/61 [04:58<03:05,  7.15s/it]

Number of matches 26268
Number of matches After Lowe's Ratio 1346
Number of Robust matches 683


 59%|[redacted]     | 36/61 [05:05<02:58,  7.16s/it]

Number of matches 31991
Number of matches After Lowe's Ratio 1731
Number of Robust matches 828


 61%|[redacted]     | 37/61 [05:15<03:07,  7.83s/it]

Number of matches 44712
Number of matches After Lowe's Ratio 1786
Number of Robust matches 709


 62%|[redacted]     | 38/61 [05:28<03:40,  9.58s/it]

Number of matches 48503
Number of matches After Lowe's Ratio 2360
Number of Robust matches 653


 64%|[redacted]     | 39/61 [05:43<04:05, 11.14s/it]

Number of matches 44596
Number of matches After Lowe's Ratio 2162
Number of Robust matches 723


 66%|[redacted]     | 40/61 [05:56<04:05, 11.69s/it]

Number of matches 35513
Number of matches After Lowe's Ratio 2137
Number of Robust matches 1064


 67%|[redacted]     | 41/61 [06:06<03:43, 11.16s/it]

Number of matches 32918
Number of matches After Lowe's Ratio 2566
Number of Robust matches 1649


 69%|[redacted]     | 42/61 [06:16<03:23, 10.71s/it]

Number of matches 30919
Number of matches After Lowe's Ratio 2632
Number of Robust matches 1697


 70%|[redacted]     | 43/61 [06:24<03:00, 10.05s/it]

Number of matches 30527
Number of matches After Lowe's Ratio 2827
```

Number of Robust matches 1832

72%|▉▉▉▉▉▉▉ | 44/61 [06:33<02:42, 9.57s/it]

Number of matches 36841
Number of matches After Lowe's Ratio 2596
Number of Robust matches 1469

74%|▉▉▉▉▉▉▉ | 45/61 [06:44<02:38, 9.93s/it]

Number of matches 39752
Number of matches After Lowe's Ratio 3123
Number of Robust matches 1906

75%|▉▉▉▉▉▉▉ | 46/61 [06:55<02:36, 10.46s/it]

Number of matches 38017
Number of matches After Lowe's Ratio 3198
Number of Robust matches 1914

77%|▉▉▉▉▉▉▉ | 47/61 [07:06<02:29, 10.64s/it]

Number of matches 39813
Number of matches After Lowe's Ratio 3063
Number of Robust matches 1766

79%|▉▉▉▉▉▉▉ | 48/61 [07:18<02:21, 10.92s/it]

Number of matches 33280
Number of matches After Lowe's Ratio 2101
Number of Robust matches 1379

80%|▉▉▉▉▉▉▉ | 49/61 [07:27<02:05, 10.43s/it]

Number of matches 31740
Number of matches After Lowe's Ratio 3662
Number of Robust matches 2520

82%|▉▉▉▉▉▉▉ | 50/61 [07:36<01:49, 9.96s/it]

Number of matches 31022
Number of matches After Lowe's Ratio 3309
Number of Robust matches 2503

84%|▉▉▉▉▉▉▉ | 51/61 [07:44<01:34, 9.40s/it]

Number of matches 27874
Number of matches After Lowe's Ratio 1813
Number of Robust matches 1126

85%|▉▉▉▉▉▉▉ | 52/61 [07:52<01:20, 8.92s/it]

Number of matches 27354
Number of matches After Lowe's Ratio 1798
Number of Robust matches 1272

```
 87%|▉▉▉▉▉▉▉▉ | 53/61 [07:59<01:07,  8.50s/it]

Number of matches 27672
Number of matches After Lowe's Ratio 2466
Number of Robust matches 1712


 89%|▉▉▉▉▉▉▉▉ | 54/61 [08:07<00:56,  8.12s/it]

Number of matches 32783
Number of matches After Lowe's Ratio 2246
Number of Robust matches 1381


 90%|▉▉▉▉▉▉▉▉▉ | 55/61 [08:16<00:50,  8.34s/it]

Number of matches 27058
Number of matches After Lowe's Ratio 2183
Number of Robust matches 1527


 92%|▉▉▉▉▉▉▉▉▉ | 56/61 [08:23<00:40,  8.20s/it]

Number of matches 29170
Number of matches After Lowe's Ratio 1911
Number of Robust matches 947


 93%|▉▉▉▉▉▉▉▉▉ | 57/61 [08:31<00:32,  8.13s/it]

Number of matches 32843
Number of matches After Lowe's Ratio 2914
Number of Robust matches 1273


 95%|▉▉▉▉▉▉▉▉▉▉ | 58/61 [08:40<00:25,  8.35s/it]

Number of matches 33022
Number of matches After Lowe's Ratio 1874
Number of Robust matches 734


 97%|▉▉▉▉▉▉▉▉▉▉ | 59/61 [08:50<00:17,  8.70s/it]

Number of matches 35941
Number of matches After Lowe's Ratio 2843
Number of Robust matches 1163


 98%|▉▉▉▉▉▉▉▉▉▉| 60/61 [09:00<00:09,  9.01s/it]
  0%|          | 0/39 [00:00<?, ?it/s]

Number of matches 25328
Number of matches After Lowe's Ratio 817
Number of Robust matches 256


  3%|▉         | 1/39 [00:06<04:10,  6.60s/it]

Number of matches 35142
Number of matches After Lowe's Ratio 1915
Number of Robust matches 1288


  5%|▉         | 2/39 [00:16<05:14,  8.50s/it]

Number of matches 28733
```

```
Number of matches After Lowe's Ratio 2339
Number of Robust matches 1717
```

```
  8%|▊         | 3/39 [00:23<04:46,  7.97s/it]
```

```
Number of matches 25875
Number of matches After Lowe's Ratio 1273
Number of Robust matches 830
```

```
 10%|▉         | 4/39 [00:30<04:28,  7.66s/it]
```

```
Number of matches 24358
Number of matches After Lowe's Ratio 574
Number of Robust matches 283
```

```
 13%|█▎        | 5/39 [00:37<04:01,  7.12s/it]
```

```
Number of matches 21252
Number of matches After Lowe's Ratio 1681
Number of Robust matches 1291
```

```
 15%|█▌        | 6/39 [00:42<03:37,  6.59s/it]
```

```
Number of matches 31075
Number of matches After Lowe's Ratio 1098
Number of Robust matches 664
```

```
 18%|█▊        | 7/39 [00:51<03:50,  7.20s/it]
```

```
Number of matches 31868
Number of matches After Lowe's Ratio 2863
Number of Robust matches 2312
```

```
 21%|██        | 8/39 [01:00<04:03,  7.85s/it]
```

```
Number of matches 33234
Number of matches After Lowe's Ratio 3196
Number of Robust matches 2549
```

```
 23%|██▎       | 9/39 [01:09<04:07,  8.25s/it]
```

```
Number of matches 29690
Number of matches After Lowe's Ratio 2525
Number of Robust matches 1862
```

```
 26%|██▌       | 10/39 [01:17<03:57,  8.18s/it]
```

```
Number of matches 32999
Number of matches After Lowe's Ratio 2626
Number of Robust matches 1856
```

```
 28%|██▊       | 11/39 [01:26<03:56,  8.44s/it]
```

```
Number of matches 33074
Number of matches After Lowe's Ratio 1885
Number of Robust matches 1161
```

31%|███ | 12/39 [01:36<03:59, 8.87s/it]

```
Number of matches 35105
Number of matches After Lowe's Ratio 2345
Number of Robust matches 1742
```

33%|███ | 13/39 [01:46<03:59, 9.22s/it]

```
Number of matches 39186
Number of matches After Lowe's Ratio 2355
Number of Robust matches 1388
```

36%|████ | 14/39 [01:57<04:06, 9.86s/it]

```
Number of matches 37695
Number of matches After Lowe's Ratio 2637
Number of Robust matches 1423
```

38%|████ | 15/39 [02:09<04:06, 10.28s/it]

```
Number of matches 40599
Number of matches After Lowe's Ratio 2833
Number of Robust matches 1534
```

41%|████ | 16/39 [02:20<04:06, 10.72s/it]

```
Number of matches 35734
Number of matches After Lowe's Ratio 2659
Number of Robust matches 1347
```

44%|████ | 17/39 [02:30<03:49, 10.44s/it]

```
Number of matches 29132
Number of matches After Lowe's Ratio 2019
Number of Robust matches 873
```

46%|████ | 18/39 [02:38<03:22, 9.65s/it]

```
Number of matches 31566
Number of matches After Lowe's Ratio 2678
Number of Robust matches 1027
```

49%|████ | 19/39 [02:47<03:08, 9.44s/it]

```
Number of matches 30048
Number of matches After Lowe's Ratio 2206
Number of Robust matches 839
```

51%|████ | 20/39 [02:55<02:50, 9.00s/it]

```
Number of matches 24007
Number of matches After Lowe's Ratio 1597
Number of Robust matches 748
```

54%|████ | 21/39 [03:01<02:27, 8.18s/it]

```
Number of matches 29055
```

Number of matches 29059
Number of matches After Lowe's Ratio 1406
Number of Robust matches 739

56%|[redacted]| 22/39 [03:09<02:18, 8.16s/it]

Number of matches 45487
Number of matches After Lowe's Ratio 545
Number of Robust matches 119

59%|[redacted]| 23/39 [03:23<02:38, 9.92s/it]

Number of matches 41464
Number of matches After Lowe's Ratio 1109
Number of Robust matches 436

62%|[redacted]| 24/39 [03:36<02:41, 10.76s/it]

Number of matches 47398
Number of matches After Lowe's Ratio 319
Number of Robust matches 6

64%|[redacted]| 25/39 [03:50<02:44, 11.78s/it]

Number of matches 36716
Number of matches After Lowe's Ratio 1045
Number of Robust matches 336

67%|[redacted]| 26/39 [04:00<02:27, 11.32s/it]

Number of matches 34503
Number of matches After Lowe's Ratio 1951
Number of Robust matches 755

69%|[redacted]| 27/39 [04:10<02:09, 10.83s/it]

Number of matches 32306
Number of matches After Lowe's Ratio 2067
Number of Robust matches 765

72%|[redacted]| 28/39 [04:19<01:52, 10.20s/it]

Number of matches 28440
Number of matches After Lowe's Ratio 1662
Number of Robust matches 611

74%|[redacted]| 29/39 [04:26<01:34, 9.42s/it]

Number of matches 26060
Number of matches After Lowe's Ratio 1387
Number of Robust matches 505

77%|[redacted]| 30/39 [04:33<01:17, 8.64s/it]

Number of matches 27600
Number of matches After Lowe's Ratio 1365
Number of Robust matches 458

```
 79%|███████▉  | 31/39 [04:41<01:06,  8.32s/it]
```

```
Number of matches 29911
Number of matches After Lowe's Ratio 2486
Number of Robust matches 859
```

```
 82%|████████  | 32/39 [04:49<00:57,  8.18s/it]
```

```
Number of matches 31772
Number of matches After Lowe's Ratio 1406
Number of Robust matches 490
```

```
 85%|████████  | 33/39 [04:58<00:50,  8.44s/it]
```

```
Number of matches 28612
Number of matches After Lowe's Ratio 2172
Number of Robust matches 984
```

```
 87%|████████▏ | 34/39 [05:06<00:41,  8.30s/it]
```

```
Number of matches 32692
Number of matches After Lowe's Ratio 1974
Number of Robust matches 780
```

```
 90%|████████▏ | 35/39 [05:14<00:33,  8.45s/it]
```

```
Number of matches 25607
Number of matches After Lowe's Ratio 1507
Number of Robust matches 753
```

```
 92%|████████▎ | 36/39 [05:21<00:23,  7.85s/it]
```

```
Number of matches 28478
Number of matches After Lowe's Ratio 1126
Number of Robust matches 778
```

```
 95%|████████▍ | 37/39 [05:29<00:15,  7.94s/it]
```

```
Number of matches 28772
Number of matches After Lowe's Ratio 1433
Number of Robust matches 1008
```

```
 97%|████████▌ | 38/39 [05:37<00:08,  8.87s/it]
```

```
Number of matches 27878
Number of matches After Lowe's Ratio 1494
Number of Robust matches 1009
```

In [94]:

```
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
_brisk[j:j+2][::-1],points_all_left_brisk[j:j+2][::-1],descriptors_all_left_brisk[j:j+2]
[::-1])
    H_left_akaze.append(H_a)
    num_matches_akaze.append(matches)
    num_good_matches_akaze.append(gd_matches)

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

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

```
  2%|          | 1/61 [00:05<05:52,  5.87s/it]
```

```
Number of matches 25071
Number of matches After Lowe's Ratio 847
Number of Robust matches 354
```

```
  3%|          | 2/61 [00:12<06:29,  6.60s/it]
```

```
Number of matches 30932
Number of matches After Lowe's Ratio 689
Number of Robust matches 238
```

```
  5%|          | 3/61 [00:21<07:06,  7.36s/it]
```

```
Number of matches 26045
Number of matches After Lowe's Ratio 322
Number of Robust matches 34
```

```
  7%|          | 4/61 [00:27<06:39,  7.01s/it]
```

```
Number of matches 23435
Number of matches After Lowe's Ratio 1570
Number of Robust matches 942
```

```
  8%|          | 5/61 [00:33<06:17,  6.74s/it]
```

```
Number of matches 28290
Number of matches After Lowe's Ratio 2016
Number of Robust matches 1122
```

```
 10%|          | 6/61 [00:41<06:33,  7.15s/it]
```

```
Number of matches 26533
Number of matches After Lowe's Ratio 1680
Number of Robust matches 869
```

```
 11%|          | 7/61 [00:48<06:22,  7.08s/it]
```

```
Number of matches 32308
Number of matches After Lowe's Ratio 2167
Number of Robust matches 1104
```

```
 13%|▉         | 8/61 [00:57<06:40,  7.55s/it]
Number of matches 22852
Number of matches After Lowe's Ratio 1054
Number of Robust matches 544

 15%|▉         | 9/61 [01:03<06:11,  7.15s/it]
Number of matches 31117
Number of matches After Lowe's Ratio 1843
Number of Robust matches 1086

 16%|▉         | 10/61 [01:11<06:21,  7.48s/it]
Number of matches 26415
Number of matches After Lowe's Ratio 1241
Number of Robust matches 713

 18%|▉         | 11/61 [01:19<06:15,  7.51s/it]
Number of matches 32660
Number of matches After Lowe's Ratio 2213
Number of Robust matches 1561

 20%|▉         | 12/61 [01:28<06:33,  8.04s/it]
Number of matches 32145
Number of matches After Lowe's Ratio 2545
Number of Robust matches 1911

 21%|▉         | 13/61 [01:37<06:40,  8.34s/it]
Number of matches 37729
Number of matches After Lowe's Ratio 2585
Number of Robust matches 1596


Number of matches 37557
Number of matches After Lowe's Ratio 3698
 23%|▉         | 14/61 [01:49<07:15,  9.27s/it]
Number of Robust matches 2416

 25%|▉         | 15/61 [01:59<07:24,  9.65s/it]
Number of matches 34855
Number of matches After Lowe's Ratio 2785
Number of Robust matches 2044

 26%|▉         | 16/61 [02:09<07:14,  9.65s/it]
Number of matches 30877
Number of matches After Lowe's Ratio 2822
Number of Robust matches 2126
```

28%|██▊ | 17/61 [02:17<06:45, 9.22s/it]

```
Number of matches 32047
Number of matches After Lowe's Ratio 2682
Number of Robust matches 1977
```

30%|███ | 18/61 [02:27<06:39, 9.30s/it]

```
Number of matches 33091
Number of matches After Lowe's Ratio 3107
Number of Robust matches 2126
```

31%|███ | 19/61 [02:36<06:28, 9.25s/it]

```
Number of matches 31279
Number of matches After Lowe's Ratio 3374
Number of Robust matches 2661
```

33%|███▎ | 20/61 [02:44<06:11, 9.06s/it]

```
Number of matches 31422
Number of matches After Lowe's Ratio 2502
Number of Robust matches 1792
```

34%|███▍ | 21/61 [02:54<06:03, 9.10s/it]

```
Number of matches 32143
Number of matches After Lowe's Ratio 2015
Number of Robust matches 1408
```

36%|███▌ | 22/61 [03:02<05:49, 8.97s/it]

```
Number of matches 31819
Number of matches After Lowe's Ratio 2231
Number of Robust matches 1447
```

38%|███▊ | 23/61 [03:11<05:38, 8.90s/it]

```
Number of matches 33063
Number of matches After Lowe's Ratio 2355
Number of Robust matches 1570
```

39%|███▉ | 24/61 [03:20<05:35, 9.06s/it]

```
Number of matches 35871
Number of matches After Lowe's Ratio 1993
Number of Robust matches 1169
```

41%|████ | 25/61 [03:31<05:45, 9.59s/it]

```
Number of matches 43925
Number of matches After Lowe's Ratio 2269
Number of Robust matches 972
```

43%|████▎ | 26/61 [03:44<06:06, 10.47s/it]

```
Number of matches 37982
```

Number of matches After Lowe's Ratio 1828
Number of Robust matches 943

44%|████ | 27/61 [03:54<05:58, 10.53s/it]

Number of matches 32060
Number of matches After Lowe's Ratio 1821
Number of Robust matches 903

46%|████ | 28/61 [04:04<05:33, 10.11s/it]

Number of matches 30283
Number of matches After Lowe's Ratio 1557
Number of Robust matches 657

48%|████ | 29/61 [04:12<05:06, 9.57s/it]

Number of matches 33737
Number of matches After Lowe's Ratio 1017
Number of Robust matches 427

49%|████ | 30/61 [04:21<04:56, 9.57s/it]

Number of matches 34231
Number of matches After Lowe's Ratio 1387
Number of Robust matches 593

51%|████ | 31/61 [04:31<04:45, 9.53s/it]

Number of matches 34536
Number of matches After Lowe's Ratio 696
Number of Robust matches 238

52%|████ | 32/61 [04:40<04:36, 9.53s/it]

Number of matches 24595
Number of matches After Lowe's Ratio 347
Number of Robust matches 41

54%|████ | 33/61 [04:47<04:02, 8.66s/it]

Number of matches 24705
Number of matches After Lowe's Ratio 1390
Number of Robust matches 642

56%|████ | 34/61 [04:53<03:35, 7.97s/it]

Number of matches 21203
Number of matches After Lowe's Ratio 1387
Number of Robust matches 761

57%|████ | 35/61 [04:59<03:05, 7.15s/it]

Number of matches 26268
Number of matches After Lowe's Ratio 1336
Number of Robust matches 773

59%|▉▉▉▉▉▉ | 36/61 [05:06<03:00, 7.22s/it]

```
Number of matches 31991
Number of matches After Lowe's Ratio 1737
Number of Robust matches 886
```

61%|▉▉▉▉▉▉ | 37/61 [05:15<03:08, 7.85s/it]

```
Number of matches 44712
Number of matches After Lowe's Ratio 1779
Number of Robust matches 726
```

62%|▉▉▉▉▉▉ | 38/61 [05:29<03:39, 9.55s/it]

```
Number of matches 48503
Number of matches After Lowe's Ratio 2342
Number of Robust matches 698
```

64%|▉▉▉▉▉▉ | 39/61 [05:44<04:04, 11.13s/it]

```
Number of matches 44596
Number of matches After Lowe's Ratio 2163
Number of Robust matches 738
```

66%|▉▉▉▉▉▉ | 40/61 [05:57<04:05, 11.70s/it]

```
Number of matches 35513
Number of matches After Lowe's Ratio 2150
Number of Robust matches 1124
```

67%|▉▉▉▉▉▉ | 41/61 [06:07<03:43, 11.17s/it]

```
Number of matches 32918
Number of matches After Lowe's Ratio 2590
Number of Robust matches 1565
```

69%|▉▉▉▉▉▉ | 42/61 [06:16<03:22, 10.68s/it]

```
Number of matches 30919
Number of matches After Lowe's Ratio 2640
Number of Robust matches 1786
```

70%|▉▉▉▉▉▉ | 43/61 [06:25<03:01, 10.07s/it]

```
Number of matches 30527
Number of matches After Lowe's Ratio 2790
Number of Robust matches 1933
```

72%|▉▉▉▉▉▉ | 44/61 [06:33<02:42, 9.58s/it]

```
Number of matches 36841
Number of matches After Lowe's Ratio 2604
Number of Robust matches 1588
```

```
Number of matches 39752
Number of matches After Lowe's Ratio 3175
```

74%|▉▉▉▉▉▉▉ | 45/61 [06:44<02:40, 10.06s/it]

Number of Robust matches 1832

75%|▉▉▉▉▉▉▉ | 46/61 [06:56<02:37, 10.47s/it]

Number of matches 38017
Number of matches After Lowe's Ratio 3160
Number of Robust matches 1904

77%|▉▉▉▉▉▉▉ | 47/61 [07:07<02:28, 10.60s/it]

Number of matches 39813
Number of matches After Lowe's Ratio 3048
Number of Robust matches 1804

79%|▉▉▉▉▉▉▉ | 48/61 [07:19<02:22, 10.99s/it]

Number of matches 33280
Number of matches After Lowe's Ratio 2111
Number of Robust matches 1364

80%|▉▉▉▉▉▉▉▉ | 49/61 [07:28<02:05, 10.45s/it]

Number of matches 31740
Number of matches After Lowe's Ratio 3645
Number of Robust matches 2591

82%|▉▉▉▉▉▉▉▉ | 50/61 [07:36<01:48, 9.84s/it]

Number of matches 31022
Number of matches After Lowe's Ratio 3270
Number of Robust matches 2519

84%|▉▉▉▉▉▉▉▉ | 51/61 [07:45<01:34, 9.47s/it]

Number of matches 27874
Number of matches After Lowe's Ratio 1775
Number of Robust matches 1054

85%|▉▉▉▉▉▉▉▉ | 52/61 [07:52<01:20, 8.92s/it]

Number of matches 27354
Number of matches After Lowe's Ratio 1767
Number of Robust matches 1196

87%|▉▉▉▉▉▉▉▉ | 53/61 [08:00<01:07, 8.40s/it]

Number of matches 27672
Number of matches After Lowe's Ratio 2471
Number of Robust matches 1849

89%|▉▉▉▉▉▉▉▉ | 54/61 [08:07<00:56, 8.04s/it]

Number of matches 32783
Number of matches After Lowe's Ratio 2259
Number of Robust matches 1213

```
 90%|████████▉ | 55/61 [08:16<00:49,  8.26s/it]
Number of matches 27058
Number of matches After Lowe's Ratio 2165
Number of Robust matches 1501

 92%|█████████▏| 56/61 [08:23<00:40,  8.11s/it]
Number of matches 29170
Number of matches After Lowe's Ratio 1882
Number of Robust matches 939

 93%|█████████▎| 57/61 [08:31<00:32,  8.01s/it]
Number of matches 32843
Number of matches After Lowe's Ratio 2901
Number of Robust matches 1354

 95%|█████████▍| 58/61 [08:40<00:24,  8.32s/it]
Number of matches 33022
Number of matches After Lowe's Ratio 1905
Number of Robust matches 741

 97%|█████████▋| 59/61 [08:50<00:17,  8.81s/it]
Number of matches 35941
Number of matches After Lowe's Ratio 2872
Number of Robust matches 1168

 98%|█████████▊| 60/61 [09:00<00:09,  9.01s/it]
  0%|          | 0/39 [00:00<?, ?it/s]
Number of matches 25328
Number of matches After Lowe's Ratio 794
Number of Robust matches 250

  3%|▎         | 1/39 [00:06<04:10,  6.59s/it]
Number of matches 35142
Number of matches After Lowe's Ratio 1913
Number of Robust matches 1359

  5%|▌         | 2/39 [00:15<05:03,  8.19s/it]
Number of matches 28733
Number of matches After Lowe's Ratio 2320
Number of Robust matches 1653

  8%|▊         | 3/39 [00:24<04:55,  8.20s/it]
Number of matches 25875
Number of matches After Lowe's Ratio 1268
Number of Robust matches 766

 10%|▉         | 4/39 [00:30<04:26,  7.61s/it]
```

10%|▉ | 4/39 [00:30<04:26, 7.61s/it]

```
Number of matches 24358
Number of matches After Lowe's Ratio 573
Number of Robust matches 289
```

13%|▌ | 5/39 [00:36<03:58, 7.00s/it]

```
Number of matches 21252
Number of matches After Lowe's Ratio 1689
Number of Robust matches 1103
```

15%|▋ | 6/39 [00:42<03:34, 6.51s/it]

```
Number of matches 31075
Number of matches After Lowe's Ratio 1105
Number of Robust matches 549
```

18%|▊ | 7/39 [00:51<03:51, 7.23s/it]

```
Number of matches 31868
Number of matches After Lowe's Ratio 2874
Number of Robust matches 2085
```

21%|▉ | 8/39 [00:59<04:01, 7.78s/it]

```
Number of matches 33234
Number of matches After Lowe's Ratio 3181
Number of Robust matches 2688
```

23%|▉ | 9/39 [01:09<04:05, 8.19s/it]

```
Number of matches 29690
Number of matches After Lowe's Ratio 2545
Number of Robust matches 1988
```

26%|█ | 10/39 [01:17<03:58, 8.24s/it]

```
Number of matches 32999
Number of matches After Lowe's Ratio 2631
Number of Robust matches 1891
```

28%|█▏ | 11/39 [01:26<04:00, 8.60s/it]

```
Number of matches 33074
Number of matches After Lowe's Ratio 1890
Number of Robust matches 1438
```

31%|█▏ | 12/39 [01:35<03:56, 8.77s/it]

```
Number of matches 35105
Number of matches After Lowe's Ratio 2299
Number of Robust matches 1596
```

33%|█▎ | 13/39 [01:46<04:00, 9.24s/it]

```
Number of matches 39186
Number of matches After Lowe's Ratio 2344
```

```
Number of Robust matches 1311

 36%|███       | 14/39 [01:57<04:05,  9.83s/it]

Number of matches 37695
Number of matches After Lowe's Ratio 2670
Number of Robust matches 1633

 38%|███       | 15/39 [02:08<04:06, 10.26s/it]

Number of matches 40599
Number of matches After Lowe's Ratio 2833
Number of Robust matches 1404

 41%|████      | 16/39 [02:20<04:06, 10.72s/it]

Number of matches 35734
Number of matches After Lowe's Ratio 2655
Number of Robust matches 1276

 44%|████      | 17/39 [02:30<03:51, 10.51s/it]

Number of matches 29132
Number of matches After Lowe's Ratio 2027
Number of Robust matches 960

 46%|████      | 18/39 [02:38<03:23,  9.69s/it]

Number of matches 31566
Number of matches After Lowe's Ratio 2633
Number of Robust matches 1133

 49%|████      | 19/39 [02:46<03:06,  9.34s/it]

Number of matches 30048
Number of matches After Lowe's Ratio 2243
Number of Robust matches 880

 51%|█████     | 20/39 [02:55<02:50,  8.99s/it]

Number of matches 24007
Number of matches After Lowe's Ratio 1628
Number of Robust matches 659

 54%|█████     | 21/39 [03:01<02:26,  8.14s/it]

Number of matches 29055
Number of matches After Lowe's Ratio 1429
Number of Robust matches 792

 56%|█████     | 22/39 [03:09<02:20,  8.25s/it]

Number of matches 45487
Number of matches After Lowe's Ratio 547
Number of Robust matches 138
```

```
 59%|██████    | 23/39 [03:23<02:39,  9.94s/it]

Number of matches 41464
Number of matches After Lowe's Ratio 1096
Number of Robust matches 443


 62%|██████    | 24/39 [03:36<02:42, 10.85s/it]

Number of matches 47398
Number of matches After Lowe's Ratio 287
Number of Robust matches 7


 64%|██████    | 25/39 [03:50<02:45, 11.79s/it]

Number of matches 36716
Number of matches After Lowe's Ratio 1065
Number of Robust matches 334


 67%|██████    | 26/39 [04:00<02:27, 11.33s/it]

Number of matches 34503
Number of matches After Lowe's Ratio 1964
Number of Robust matches 713


 69%|██████    | 27/39 [04:10<02:11, 10.95s/it]

Number of matches 32306
Number of matches After Lowe's Ratio 2061
Number of Robust matches 745


 72%|███████   | 28/39 [04:19<01:53, 10.29s/it]

Number of matches 28440
Number of matches After Lowe's Ratio 1679
Number of Robust matches 656


 74%|███████   | 29/39 [04:27<01:34,  9.46s/it]

Number of matches 26060
Number of matches After Lowe's Ratio 1311
Number of Robust matches 444


 77%|███████   | 30/39 [04:34<01:18,  8.69s/it]

Number of matches 27600
Number of matches After Lowe's Ratio 1328
Number of Robust matches 435


 79%|███████   | 31/39 [04:41<01:07,  8.44s/it]

Number of matches 29911
Number of matches After Lowe's Ratio 2525
Number of Robust matches 868


 82%|████████  | 32/39 [04:49<00:58,  8.29s/it]

Number of matches 31772
Number of matches After Lowe's Ratio 1433
```

```
Number of Robust matches 484
```

```
 85%|████████  | 33/39 [04:58<00:50,  8.49s/it]
```

```
Number of matches 28612
Number of matches After Lowe's Ratio 2189
Number of Robust matches 1070
```

```
 87%|████████  | 34/39 [05:06<00:41,  8.32s/it]
```

```
Number of matches 32692
Number of matches After Lowe's Ratio 2002
Number of Robust matches 853
```

```
 90%|████████  | 35/39 [05:15<00:34,  8.60s/it]
```

```
Number of matches 25607
Number of matches After Lowe's Ratio 1511
Number of Robust matches 702
```

```
 92%|████████  | 36/39 [05:22<00:24,  8.04s/it]
```

```
Number of matches 28478
Number of matches After Lowe's Ratio 1119
Number of Robust matches 809
```

```
 95%|████████  | 37/39 [05:30<00:15,  7.89s/it]
```

```
Number of matches 28772
Number of matches After Lowe's Ratio 1474
Number of Robust matches 1108
```

```
 97%|████████  | 38/39 [05:37<00:08,  8.89s/it]
```

```
Number of matches 27878
Number of matches After Lowe's Ratio 1490
Number of Robust matches 998
```

In [95]:

```
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
_brisk[j:j+2][::-1],points_all_left_brisk[j:j+2][::-1],descriptors_all_left_brisk[j:j+2]
[::-1])
    H_left_kaze.append(H_a)
    num_matches_kaze.append(matches)
    num_good_matches_kaze.append(gd_matches)

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

```
        break

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

```
  2%|▏          | 1/61 [00:06<06:04,  6.08s/it]
```

```
Number of matches 25071
Number of matches After Lowe's Ratio 842
Number of Robust matches 321
```

```
  3%|▏          | 2/61 [00:13<06:33,  6.67s/it]
```

```
Number of matches 30932
Number of matches After Lowe's Ratio 714
Number of Robust matches 268
```

```
  5%|▎          | 3/61 [00:21<07:08,  7.39s/it]
```

```
Number of matches 26045
Number of matches After Lowe's Ratio 331
Number of Robust matches 29
```

```
  7%|▍          | 4/61 [00:28<06:47,  7.15s/it]
```

```
Number of matches 23435
Number of matches After Lowe's Ratio 1597
Number of Robust matches 981
```

```
  8%|▍          | 5/61 [00:34<06:13,  6.68s/it]
```

```
Number of matches 28290
Number of matches After Lowe's Ratio 1978
Number of Robust matches 1044
```

```
 10%|▌          | 6/61 [00:41<06:28,  7.06s/it]
```

```
Number of matches 26533
Number of matches After Lowe's Ratio 1716
Number of Robust matches 870
```

```
 11%|▋          | 7/61 [00:49<06:31,  7.24s/it]
```

```
Number of matches 32308
Number of matches After Lowe's Ratio 2245
Number of Robust matches 1280
```

```
 13%|▊          | 8/61 [00:57<06:44,  7.63s/it]
```

```
Number of matches 22852
Number of matches After Lowe's Ratio 1020
Number of Robust matches 537
```

```
 15%|▉          | 9/61 [01:03<06:08,  7.08s/it]
```

```
Number of matches 31117
```

Number of matches 3111?
Number of matches After Lowe's Ratio 1793
Number of Robust matches 1269

16%|▉ | 10/61 [01:12<06:24, 7.54s/it]

Number of matches 26415
Number of matches After Lowe's Ratio 1216
Number of Robust matches 797

18%|▉ | 11/61 [01:19<06:15, 7.52s/it]

Number of matches 32660
Number of matches After Lowe's Ratio 2185
Number of Robust matches 1572

20%|▉ | 12/61 [01:28<06:27, 7.92s/it]

Number of matches 32145
Number of matches After Lowe's Ratio 2496
Number of Robust matches 1697

21%|▉ | 13/61 [01:37<06:38, 8.31s/it]

Number of matches 37729
Number of matches After Lowe's Ratio 2542
Number of Robust matches 1679

23%|▉ | 14/61 [01:49<07:12, 9.20s/it]

Number of matches 37557
Number of matches After Lowe's Ratio 3713
Number of Robust matches 2969

25%|▉ | 15/61 [01:59<07:23, 9.63s/it]

Number of matches 34855
Number of matches After Lowe's Ratio 2835
Number of Robust matches 2213

26%|▉ | 16/61 [02:09<07:16, 9.70s/it]

Number of matches 30877
Number of matches After Lowe's Ratio 2838
Number of Robust matches 2267

28%|▉ | 17/61 [02:18<06:49, 9.31s/it]

Number of matches 32047
Number of matches After Lowe's Ratio 2648
Number of Robust matches 1712

30%|▉ | 18/61 [02:27<06:40, 9.30s/it]

Number of matches 33091
Number of matches After Lowe's Ratio 3111
Number of Robust matches 2509

31%|███ | 19/61 [02:36<06:29, 9.28s/it]

```
Number of matches 31279
Number of matches After Lowe's Ratio 3372
Number of Robust matches 2261
```

33%|███ | 20/61 [02:45<06:10, 9.05s/it]

```
Number of matches 31422
Number of matches After Lowe's Ratio 2540
Number of Robust matches 1905
```

34%|███ | 21/61 [02:54<06:02, 9.05s/it]

```
Number of matches 32143
Number of matches After Lowe's Ratio 2002
Number of Robust matches 1337
```

36%|███ | 22/61 [03:03<05:53, 9.07s/it]

```
Number of matches 31819
Number of matches After Lowe's Ratio 2218
Number of Robust matches 1425
```

38%|███ | 23/61 [03:12<05:42, 9.02s/it]

```
Number of matches 33063
Number of matches After Lowe's Ratio 2355
Number of Robust matches 1583
```

39%|███ | 24/61 [03:21<05:36, 9.10s/it]

```
Number of matches 35871
Number of matches After Lowe's Ratio 2030
Number of Robust matches 1296
```

41%|████ | 25/61 [03:32<05:47, 9.65s/it]

```
Number of matches 43925
Number of matches After Lowe's Ratio 2292
Number of Robust matches 943
```

43%|████ | 26/61 [03:45<06:10, 10.57s/it]

```
Number of matches 37982
Number of matches After Lowe's Ratio 1836
Number of Robust matches 911
```

44%|████ | 27/61 [03:56<06:05, 10.74s/it]

```
Number of matches 32060
Number of matches After Lowe's Ratio 1850
Number of Robust matches 1062
```

46%|████ | 28/61 [04:04<05:33, 10.10s/it]

```
Number of matches 30283
Number of matches After Lowe's Ratio 1596
Number of Robust matches 647


 48%|████      | 29/61 [04:13<05:05,  9.56s/it]

Number of matches 33737
Number of matches After Lowe's Ratio 1004
Number of Robust matches 382


 49%|████      | 30/61 [04:22<04:55,  9.53s/it]

Number of matches 34231
Number of matches After Lowe's Ratio 1384
Number of Robust matches 589


 51%|█████     | 31/61 [04:32<04:51,  9.72s/it]

Number of matches 34536
Number of matches After Lowe's Ratio 657
Number of Robust matches 248


 52%|█████     | 32/61 [04:42<04:40,  9.68s/it]

Number of matches 24595
Number of matches After Lowe's Ratio 371
Number of Robust matches 49


 54%|█████     | 33/61 [04:48<04:01,  8.62s/it]

Number of matches 24705
Number of matches After Lowe's Ratio 1383
Number of Robust matches 699


 56%|█████     | 34/61 [04:54<03:33,  7.92s/it]

Number of matches 21203
Number of matches After Lowe's Ratio 1403
Number of Robust matches 793


 57%|█████     | 35/61 [05:00<03:10,  7.34s/it]

Number of matches 26268
Number of matches After Lowe's Ratio 1328
Number of Robust matches 678


 59%|█████     | 36/61 [05:07<03:02,  7.29s/it]

Number of matches 31991
Number of matches After Lowe's Ratio 1790
Number of Robust matches 945


 61%|██████    | 37/61 [05:17<03:08,  7.86s/it]

Number of matches 44712
Number of matches After Lowe's Ratio 1763
Number of Robust matches 701
```

62%|██████ | 38/61 [05:30<03:38, 9.51s/it]

```
Number of matches 48503
Number of matches After Lowe's Ratio 2344
Number of Robust matches 667
```

64%|██████ | 39/61 [05:45<04:07, 11.27s/it]

```
Number of matches 44596
Number of matches After Lowe's Ratio 2165
Number of Robust matches 860
```

66%|██████ | 40/61 [05:58<04:05, 11.71s/it]

```
Number of matches 35513
Number of matches After Lowe's Ratio 2175
Number of Robust matches 1086
```

67%|██████ | 41/61 [06:08<03:46, 11.31s/it]

```
Number of matches 32918
Number of matches After Lowe's Ratio 2628
Number of Robust matches 1696
```

69%|██████ | 42/61 [06:18<03:22, 10.68s/it]

```
Number of matches 30919
Number of matches After Lowe's Ratio 2594
Number of Robust matches 1802
```

70%|███████ | 43/61 [06:26<02:58, 9.93s/it]

```
Number of matches 30527
Number of matches After Lowe's Ratio 2840
Number of Robust matches 1958
```

72%|███████ | 44/61 [06:34<02:41, 9.49s/it]

```
Number of matches 36841
Number of matches After Lowe's Ratio 2591
Number of Robust matches 1592
```

74%|███████ | 45/61 [06:46<02:40, 10.03s/it]

```
Number of matches 39752
Number of matches After Lowe's Ratio 3176
Number of Robust matches 1941
```

75%|███████ | 46/61 [06:57<02:36, 10.42s/it]

```
Number of matches 38017
Number of matches After Lowe's Ratio 3164
Number of Robust matches 1786
```

77%|███████ | 47/61 [07:08<02:28, 10.59s/it]

```
Number of matches 39813
Number of matches After Lowe's Ratio 3073
Number of Robust matches 1753


 79%|███████▉  | 48/61 [07:20<02:22, 10.99s/it]

Number of matches 33280
Number of matches After Lowe's Ratio 2082
Number of Robust matches 1260


 80%|████████  | 49/61 [07:29<02:05, 10.44s/it]

Number of matches 31740
Number of matches After Lowe's Ratio 3643
Number of Robust matches 2667


 82%|████████▏ | 50/61 [07:38<01:48,  9.91s/it]

Number of matches 31022
Number of matches After Lowe's Ratio 3261
Number of Robust matches 2529


 84%|████████▎ | 51/61 [07:47<01:36,  9.64s/it]

Number of matches 27874
Number of matches After Lowe's Ratio 1830
Number of Robust matches 1121


 85%|████████▌ | 52/61 [07:54<01:19,  8.88s/it]

Number of matches 27354
Number of matches After Lowe's Ratio 1760
Number of Robust matches 1280


 87%|████████▋ | 53/61 [08:01<01:06,  8.37s/it]

Number of matches 27672
Number of matches After Lowe's Ratio 2408
Number of Robust matches 1805


 89%|████████▊ | 54/61 [08:09<00:57,  8.16s/it]

Number of matches 32783
Number of matches After Lowe's Ratio 2218
Number of Robust matches 1336


 90%|█████████ | 55/61 [08:18<00:51,  8.51s/it]

Number of matches 27058
Number of matches After Lowe's Ratio 2172
Number of Robust matches 1511


 92%|█████████▏| 56/61 [08:25<00:40,  8.08s/it]

Number of matches 29170
Number of matches After Lowe's Ratio 1888
Number of Robust matches 885
```

```
 93%|█████████ | 57/61 [08:33<00:32,  8.10s/it]
Number of matches 32843
Number of matches After Lowe's Ratio 2846
Number of Robust matches 1460

 95%|██████████ | 58/61 [08:42<00:25,  8.45s/it]
Number of matches 33022
Number of matches After Lowe's Ratio 1936
Number of Robust matches 822

 97%|██████████ | 59/61 [08:52<00:17,  8.80s/it]
Number of matches 35941
Number of matches After Lowe's Ratio 2847
Number of Robust matches 1139

 98%|██████████ | 60/61 [09:02<00:09,  9.04s/it]
  0%|           | 0/39 [00:00<?, ?it/s]
Number of matches 25328
Number of matches After Lowe's Ratio 800
Number of Robust matches 265

  3%|▎          | 1/39 [00:06<04:11,  6.62s/it]
Number of matches 35142
Number of matches After Lowe's Ratio 1892
Number of Robust matches 1170

  5%|▌          | 2/39 [00:16<05:09,  8.38s/it]
Number of matches 28733
Number of matches After Lowe's Ratio 2334
Number of Robust matches 1830

  8%|▊          | 3/39 [00:24<04:56,  8.23s/it]
Number of matches 25875
Number of matches After Lowe's Ratio 1252
Number of Robust matches 804

 10%|█          | 4/39 [00:30<04:23,  7.54s/it]
Number of matches 24358
Number of matches After Lowe's Ratio 580
Number of Robust matches 269

 13%|█▎         | 5/39 [00:37<04:01,  7.11s/it]
Number of matches 21252
Number of matches After Lowe's Ratio 1705
Number of Robust matches 1136
```

```
 15%|▉         | 6/39 [00:42<03:37,  6.59s/it]

Number of matches 31075
Number of matches After Lowe's Ratio 1100
Number of Robust matches 605


 18%|▉         | 7/39 [00:51<03:51,  7.22s/it]

Number of matches 31868
Number of matches After Lowe's Ratio 2854
Number of Robust matches 2157



Number of matches 33234
Number of matches After Lowe's Ratio 3166
 21%|▉         | 8/39 [01:00<04:06,  7.97s/it]

Number of Robust matches 2598


 23%|▉         | 9/39 [01:09<04:09,  8.30s/it]

Number of matches 29690
Number of matches After Lowe's Ratio 2541
Number of Robust matches 2137


 26%|▉         | 10/39 [01:17<03:59,  8.26s/it]

Number of matches 32999
Number of matches After Lowe's Ratio 2670
Number of Robust matches 2144


 28%|▉         | 11/39 [01:27<03:59,  8.54s/it]

Number of matches 33074
Number of matches After Lowe's Ratio 1880
Number of Robust matches 1348


 31%|▉         | 12/39 [01:36<03:58,  8.82s/it]

Number of matches 35105
Number of matches After Lowe's Ratio 2325
Number of Robust matches 1496


 33%|▉         | 13/39 [01:46<03:58,  9.16s/it]

Number of matches 39186
Number of matches After Lowe's Ratio 2370
Number of Robust matches 1485


 36%|▉         | 14/39 [01:57<04:05,  9.83s/it]

Number of matches 37695
Number of matches After Lowe's Ratio 2630
Number of Robust matches 1540


 38%|▉         | 15/39 [02:09<04:07, 10.31s/it]

Number of matches 40599
```

```
Number of matches After Lowe's Ratio 2865
Number of Robust matches 1489


 41%|████      | 16/39 [02:20<04:05, 10.68s/it]

Number of matches 35734
Number of matches After Lowe's Ratio 2666
Number of Robust matches 1177


 44%|████▍     | 17/39 [02:30<03:50, 10.48s/it]

Number of matches 29132
Number of matches After Lowe's Ratio 2050
Number of Robust matches 983


 46%|████▊     | 18/39 [02:39<03:26,  9.85s/it]

Number of matches 31566
Number of matches After Lowe's Ratio 2611
Number of Robust matches 1069


 49%|█████     | 19/39 [02:47<03:09,  9.48s/it]

Number of matches 30048
Number of matches After Lowe's Ratio 2257
Number of Robust matches 784


 51%|█████▎    | 20/39 [02:55<02:51,  9.00s/it]

Number of matches 24007
Number of matches After Lowe's Ratio 1609
Number of Robust matches 775


 54%|█████▋    | 21/39 [03:02<02:27,  8.21s/it]

Number of matches 29055
Number of matches After Lowe's Ratio 1443
Number of Robust matches 698


 56%|█████▉    | 22/39 [03:10<02:22,  8.38s/it]

Number of matches 45487
Number of matches After Lowe's Ratio 564
Number of Robust matches 133


 59%|██████▏   | 23/39 [03:24<02:39,  9.96s/it]

Number of matches 41464
Number of matches After Lowe's Ratio 1119
Number of Robust matches 445


 62%|██████▌   | 24/39 [03:37<02:41, 10.78s/it]

Number of matches 47398
Number of matches After Lowe's Ratio 291
Number of Robust matches 6
```

64%|██████ | 25/39 [03:51<02:44, 11.78s/it]

```
Number of matches 36716
Number of matches After Lowe's Ratio 1043
Number of Robust matches 315
```

67%|██████▋ | 26/39 [04:01<02:28, 11.39s/it]

```
Number of matches 34503
Number of matches After Lowe's Ratio 1954
Number of Robust matches 815
```

69%|██████▉ | 27/39 [04:11<02:10, 10.88s/it]

```
Number of matches 32306
Number of matches After Lowe's Ratio 2086
Number of Robust matches 706
```

72%|███████▏ | 28/39 [04:20<01:52, 10.23s/it]

```
Number of matches 28440
Number of matches After Lowe's Ratio 1702
Number of Robust matches 652
```

74%|███████▍ | 29/39 [04:27<01:34, 9.44s/it]

```
Number of matches 26060
Number of matches After Lowe's Ratio 1361
Number of Robust matches 423
```

77%|███████▋ | 30/39 [04:34<01:18, 8.74s/it]

```
Number of matches 27600
Number of matches After Lowe's Ratio 1356
Number of Robust matches 375
```

79%|███████▉ | 31/39 [04:42<01:06, 8.28s/it]

```
Number of matches 29911
Number of matches After Lowe's Ratio 2452
Number of Robust matches 938
```

82%|████████▏ | 32/39 [04:50<00:58, 8.34s/it]

```
Number of matches 31772
Number of matches After Lowe's Ratio 1415
Number of Robust matches 443
```

85%|████████▍ | 33/39 [04:59<00:50, 8.43s/it]

```
Number of matches 28612
Number of matches After Lowe's Ratio 2192
Number of Robust matches 901
```

87%|████████▋ | 34/39 [05:07<00:41, 8.27s/it]

```
Number of matches 32692
```

```
Number of matches After Lowe's Ratio 1981
Number of Robust matches 861
```

```
 90%|████████▉ | 35/39 [05:16<00:34,  8.52s/it]
```

```
Number of matches 25607
Number of matches After Lowe's Ratio 1491
Number of Robust matches 767
```

```
 92%|█████████▏ | 36/39 [05:23<00:24,  8.06s/it]
```

```
Number of matches 28478
Number of matches After Lowe's Ratio 1119
Number of Robust matches 723
```

```
 95%|█████████▍ | 37/39 [05:31<00:15,  7.97s/it]
```

```
Number of matches 28772
Number of matches After Lowe's Ratio 1470
Number of Robust matches 1085
```

```
 97%|█████████▋ | 38/39 [05:38<00:08,  8.91s/it]
```

```
Number of matches 27878
Number of matches After Lowe's Ratio 1446
Number of Robust matches 929
```

In [96]:

```
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 [97]:

```
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 [98]:

```
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 [ ]:

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

In [ ]:

```
warp_imgs_all_brisk = final_steps_right_union(warp_imgs_left,images_right_bgr_no_enhance,
H_right_brisk,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

In [ ]:

```
plt.imshow(warp_imgs_all_brisk)
plt.title(' Mosaic using Brisk Image')
```

In [ ]:

```
omax,omin,umax,umin,T,H,W,HT = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_orb,H_right_orb)
```

In [ ]:

```
warp_img_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_orb,omax,omin,um
ax,umin,T,H,W,HT)
```

In [ ]:

```
warp_imgs_all_orb = final_steps_right_union(warp_img_left,images_right_bgr_no_enhance,H_r
ight_orb,omax,omin,umax,umin,T,H,W,HT)
```

In [ ]:

```
plt.imshow(warp_imgs_all_orb)
plt.title('Mosaic using Orb')
```

In [ ]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_akaze,H_right_akaze)
```

In [ ]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_akaze,amax,ami
n,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
warp_imgs_all_akaze = final_steps_right_union(warp_image_left,images_right_bgr_no_enhance
,H_right_akaze,amax,amin,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
plt.imshow(warp_imgs_all_akaze)
plt.title('Mosaic using Akaze image')
```