In [3]:

```
!pip install torchsummary
```

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

In [4]:

```
import numpy as np

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

In [5]:

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

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

```
        ratio = minval1/minval2

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

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

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

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

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

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

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

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

In [1]:

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

```
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 [2]:

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

In [6]:

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

folder_path = '../input/uni-campus-dataset/RGB-img/img/'
left_files_path_rev = []
right_files_path = []
for file in files_all[:51]:
    left_files_path_rev.append(folder_path + file)


left_files_path = left_files_path_rev[::-1]


for file in files_all[50:100]:
    right_files_path.append(folder_path + file)
```

In [7]:

```
gridsize = 8
clahe = cv2.createCLAHE(clipLimit=2.0,tileGridSize=(gridsize,gridsize))
images_left_bgr = []
images_right_bgr = []
images_left = []
images_right = []


for file in tqdm(left_files_path):
    left_image_sat= cv2.imread(file)
    lab = cv2.cvtColor(left_image_sat, cv2.COLOR_BGR2LAB)
    lab[...,0] = clahe.apply(lab[...,0])
    left_image_sat = cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)
    left_img = cv2.resize(left_image_sat,None,fx=0.35, fy=0.35, interpolation = cv2.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%|██████████| 51/51 [00:56<00:00,  1.10s/it]
100%|██████████| 50/50 [00:53<00:00,  1.08s/it]
```

In [8]:

```
images_left_bgr_no_enhance = []
images_right_bgr_no_enhance = []

for file in tqdm(left_files_path):
    left_image_sat= cv2.imread(file)
    left_img = cv2.resize(left_image_sat,None,fx=0.35, fy=0.35, interpolation = cv2.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%|██████████| 51/51 [00:20<00:00,  2.44it/s]
100%|██████████| 50/50 [00:19<00:00,  2.52it/s]
```

In [ ]:

```
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]))
```

In [ ]:

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

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

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

for imgs in tqdm(images_right_bgr):
    kpt = orb.detect(imgs,None)
```

```
    kpt,descrip = orb.compute(imgs, kpt)
    keypoints_all_right_orb.append(kpt)
    descriptors_all_right_orb.append(descrip)
    points_all_right_orb.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

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

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

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

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

In [9]:

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

In [10]:

```
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%|██████████| 51/51 [01:15<00:00,  1.49s/it]
100%|██████████| 50/50 [01:10<00:00,  1.41s/it]
```

In [ ]:

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


keypoints_all_right_star = []
```

```
descriptors_all_right_brief = []
points_all_right_star=[]

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

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

In [9]:

```
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]))
```

```
100%|██████████| 51/51 [00:46<00:00,  1.10it/s]
100%|██████████| 50/50 [00:41<00:00,  1.21it/s]
```

In [ ]:

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

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


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

```
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_mser.append(kpt)
    descriptors_all_right_mser.append(descrip)
    points_all_right_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [10]:

```
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]))
```

```
100%|██████████| 51/51 [07:00<00:00,  8.25s/it]
100%|██████████| 50/50 [07:01<00:00,  8.42s/it]
```

In [9]:

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

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

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

```
100%|██████████| 51/51 [06:10<00:00,  7.26s/it]
100%|██████████| 50/50 [06:16<00:00,  7.53s/it]
```

In [ ]:

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

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

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

In [ ]:

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

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

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

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

In [ ]:

```
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 [ ]:

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

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

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

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

In [ ]:

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

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

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

In [ ]:

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

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

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

In [10]:

```
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 [11]:

```
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]))
```

```
100%|██████████| 51/51 [02:30<00:00,  2.96s/it]
100%|██████████| 50/50 [02:30<00:00,  3.01s/it]
```

In [10]:

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

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

In [11]:

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

In [12]:

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

In [13]:

```
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 [14]:

```
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 [17]:

```
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%|          | 0/51 [00:00<?, ?it/s]
```

```
Number of pts selected: 54477
```

```
/opt/conda/lib/python3.7/site-packages/torch/nn/functional.py:3385: UserWarning: Default
grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Plea
se specify align_corners=True if the old behavior is desired. See the documentation of gr
id_sample for details.
  warnings.warn("Default grid_sample and affine_grid behavior has changed "
  2%|▏         | 1/51 [00:01<01:17,  1.55s/it]
```

```
Number of pts selected: 64559
```

```
  4%|▍         | 2/51 [00:02<00:48,  1.01it/s]
```

```
Number of pts selected: 62731
```

```
  6%|▌         | 3/51 [00:02<00:38,  1.26it/s]
```

```
Number of pts selected: 66915
  8%|▊         | 4/51 [00:03<00:33,  1.41it/s]
Number of pts selected: 65977
 10%|▉         | 5/51 [00:03<00:30,  1.51it/s]
Number of pts selected: 62256
 12%|█         | 6/51 [00:04<00:28,  1.61it/s]
Number of pts selected: 58441
 14%|█▏        | 7/51 [00:04<00:25,  1.70it/s]
Number of pts selected: 52856
 16%|█▎        | 8/51 [00:05<00:24,  1.79it/s]
Number of pts selected: 53728
 18%|█▍        | 9/51 [00:05<00:22,  1.85it/s]
Number of pts selected: 52870
 20%|█▌        | 10/51 [00:06<00:21,  1.90it/s]
Number of pts selected: 48261
 22%|█▋        | 11/51 [00:06<00:20,  1.97it/s]
Number of pts selected: 53550
 24%|█▊        | 12/51 [00:07<00:19,  1.98it/s]
Number of pts selected: 53340
 25%|█▉        | 13/51 [00:07<00:19,  1.99it/s]
Number of pts selected: 55541
 27%|██        | 14/51 [00:08<00:18,  1.98it/s]
Number of pts selected: 53723
 29%|██▏       | 15/51 [00:08<00:18,  2.00it/s]
Number of pts selected: 63642
 31%|██▎       | 16/51 [00:09<00:19,  1.76it/s]
Number of pts selected: 63850
 33%|██▍       | 17/51 [00:10<00:19,  1.77it/s]
Number of pts selected: 63381
 35%|██▋       | 18/51 [00:10<00:18,  1.78it/s]
Number of pts selected: 57585
 37%|██▊       | 19/51 [00:11<00:17,  1.82it/s]
Number of pts selected: 45309
 39%|██▉       | 20/51 [00:11<00:16,  1.93it/s]
Number of pts selected: 43137
 41%|███       | 21/51 [00:12<00:14,  2.02it/s]
Number of pts selected: 32917
 43%|███▏      | 22/51 [00:12<00:13,  2.18it/s]
Number of pts selected: 36228
 45%|███▎      | 23/51 [00:12<00:12,  2.27it/s]
```

```
Number of pts selected: 36082
 47%|████      | 24/51 [00:13<00:11,  2.35it/s]
Number of pts selected: 35452
 49%|████      | 25/51 [00:13<00:10,  2.41it/s]
Number of pts selected: 41594
 51%|█████     | 26/51 [00:14<00:10,  2.39it/s]
Number of pts selected: 41845
 53%|█████     | 27/51 [00:14<00:10,  2.37it/s]
Number of pts selected: 52565
 55%|█████     | 28/51 [00:15<00:10,  2.26it/s]
Number of pts selected: 51583
 57%|█████     | 29/51 [00:15<00:10,  2.18it/s]
Number of pts selected: 53865
 59%|█████     | 30/51 [00:16<00:09,  2.11it/s]
Number of pts selected: 51580
 61%|██████    | 31/51 [00:16<00:09,  2.10it/s]
Number of pts selected: 52497
 63%|██████    | 32/51 [00:17<00:09,  2.08it/s]
Number of pts selected: 53351
 65%|██████    | 33/51 [00:17<00:08,  2.06it/s]
Number of pts selected: 58937
 67%|██████    | 34/51 [00:18<00:08,  2.01it/s]
Number of pts selected: 63670
 69%|██████    | 35/51 [00:18<00:08,  1.95it/s]
Number of pts selected: 63164
 71%|███████   | 36/51 [00:19<00:07,  1.90it/s]
Number of pts selected: 61341
 73%|███████   | 37/51 [00:19<00:07,  1.89it/s]
Number of pts selected: 62577
 75%|███████   | 38/51 [00:20<00:07,  1.73it/s]
Number of pts selected: 60963
 76%|███████   | 39/51 [00:20<00:06,  1.76it/s]
Number of pts selected: 59765
 78%|███████   | 40/51 [00:21<00:06,  1.79it/s]
Number of pts selected: 59418
 80%|████████  | 41/51 [00:21<00:05,  1.82it/s]
Number of pts selected: 57060
 82%|████████  | 42/51 [00:22<00:05,  1.66it/s]
Number of pts selected: 60042
 84%|████████  | 43/51 [00:23<00:04,  1.60it/s]
```

```
Number of pts selected: 60481
 86%|████████▌ | 44/51 [00:23<00:04,  1.67it/s]
Number of pts selected: 61320
 88%|████████▊ | 45/51 [00:24<00:03,  1.73it/s]
Number of pts selected: 60846
 90%|████████▉ | 46/51 [00:24<00:02,  1.77it/s]
Number of pts selected: 57839
 92%|█████████▏| 47/51 [00:25<00:02,  1.81it/s]
Number of pts selected: 59702
 94%|█████████▍| 48/51 [00:26<00:01,  1.82it/s]
Number of pts selected: 58094
 96%|█████████▌| 49/51 [00:26<00:01,  1.85it/s]
Number of pts selected: 59875
 98%|█████████▊| 50/51 [00:27<00:00,  1.87it/s]
Number of pts selected: 57318
100%|██████████| 51/51 [00:27<00:00,  1.85it/s]
  0%|          | 0/50 [00:00<?, ?it/s]
Number of pts selected: 54477
  2%|▏         | 1/50 [00:00<00:24,  2.00it/s]
Number of pts selected: 56486
  4%|▍         | 2/50 [00:01<00:24,  1.99it/s]
Number of pts selected: 43862
  6%|▌         | 3/50 [00:01<00:22,  2.14it/s]
Number of pts selected: 50000
  8%|▊         | 4/50 [00:01<00:21,  2.15it/s]
Number of pts selected: 43516
 10%|▉         | 5/50 [00:02<00:20,  2.21it/s]
Number of pts selected: 41297
 12%|█▏        | 6/50 [00:02<00:19,  2.27it/s]
Number of pts selected: 42597
 14%|█▍        | 7/50 [00:03<00:18,  2.31it/s]
Number of pts selected: 44806
 16%|█▌        | 8/50 [00:03<00:20,  2.06it/s]
Number of pts selected: 43315
 18%|█▊        | 9/50 [00:04<00:19,  2.13it/s]
Number of pts selected: 39997
 20%|██        | 10/50 [00:04<00:17,  2.23it/s]
Number of pts selected: 37697
 22%|██▏       | 11/50 [00:04<00:16,  2.32it/s]
Number of pts selected: 41027
```

24%|██ | 12/50 [00:05<00:16, 2.36it/s]

Number of pts selected: 45908

26%|██ | 13/50 [00:05<00:15, 2.34it/s]

Number of pts selected: 49188

28%|██ | 14/50 [00:06<00:15, 2.30it/s]

Number of pts selected: 57475

30%|███ | 15/50 [00:06<00:15, 2.20it/s]

Number of pts selected: 55483

32%|███ | 16/50 [00:07<00:15, 2.15it/s]

Number of pts selected: 52557

34%|███ | 17/50 [00:07<00:15, 2.14it/s]

Number of pts selected: 55422

36%|███ | 18/50 [00:08<00:15, 2.11it/s]

Number of pts selected: 54316

38%|███ | 19/50 [00:08<00:15, 2.06it/s]

Number of pts selected: 51662

40%|████ | 20/50 [00:09<00:14, 2.08it/s]

Number of pts selected: 50162

42%|████ | 21/50 [00:09<00:13, 2.10it/s]

Number of pts selected: 48664

44%|████ | 22/50 [00:10<00:13, 2.14it/s]

Number of pts selected: 46023

46%|████ | 23/50 [00:10<00:12, 2.19it/s]

Number of pts selected: 48049

48%|████ | 24/50 [00:11<00:11, 2.20it/s]

Number of pts selected: 57650

50%|█████ | 25/50 [00:11<00:11, 2.13it/s]

Number of pts selected: 70799

52%|█████ | 26/50 [00:12<00:12, 1.97it/s]

Number of pts selected: 72440

54%|█████ | 27/50 [00:12<00:12, 1.88it/s]

Number of pts selected: 75195

56%|█████ | 28/50 [00:13<00:12, 1.80it/s]

Number of pts selected: 72197

58%|█████ | 29/50 [00:13<00:11, 1.77it/s]

Number of pts selected: 66162

60%|██████ | 30/50 [00:14<00:11, 1.73it/s]

Number of pts selected: 59430

62%|██████ | 31/50 [00:15<00:11, 1.70it/s]

Number of pts selected: 61074

64%|██████▍ | 32/50 [00:15<00:10, 1.75it/s]

```
Number of pts selected: 50121
```

66%|██████▌ | 33/50 [00:16<00:09, 1.85it/s]

```
Number of pts selected: 33851
```

68%|██████▊ | 34/50 [00:16<00:07, 2.05it/s]

```
Number of pts selected: 32768
```

70%|██████▉ | 35/50 [00:16<00:06, 2.21it/s]

```
Number of pts selected: 36270
```

72%|███████▏ | 36/50 [00:17<00:06, 2.28it/s]

```
Number of pts selected: 50163
```

74%|███████▍ | 37/50 [00:17<00:05, 2.24it/s]

```
Number of pts selected: 47558
```

76%|███████▌ | 38/50 [00:18<00:05, 2.24it/s]

```
Number of pts selected: 54703
```

78%|███████▊ | 39/50 [00:18<00:05, 2.16it/s]

```
Number of pts selected: 60403
```

80%|████████ | 40/50 [00:19<00:04, 2.09it/s]

```
Number of pts selected: 73839
```

82%|████████▏ | 41/50 [00:19<00:04, 1.95it/s]

```
Number of pts selected: 72110
```

84%|████████▍ | 42/50 [00:20<00:04, 1.86it/s]

```
Number of pts selected: 70241
```

86%|████████▌ | 43/50 [00:20<00:03, 1.80it/s]

```
Number of pts selected: 65699
```

88%|████████▊ | 44/50 [00:21<00:03, 1.78it/s]

```
Number of pts selected: 63686
```

90%|█████████ | 45/50 [00:22<00:02, 1.80it/s]

```
Number of pts selected: 57061
```

92%|█████████▏ | 46/50 [00:22<00:02, 1.86it/s]

```
Number of pts selected: 44654
```

94%|█████████▍ | 47/50 [00:23<00:01, 1.97it/s]

```
Number of pts selected: 42427
```

96%|█████████▌ | 48/50 [00:23<00:00, 2.08it/s]

```
Number of pts selected: 41848
```

98%|█████████▊ | 49/50 [00:23<00:00, 2.17it/s]

```
Number of pts selected: 43600
```

100%|██████████| 50/50 [00:24<00:00, 2.06it/s]

In [19]:

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

```
100%|██████████| 101/101 [00:00<00:00, 452590.50it/s]
```

In [ ]:

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

In [ ]:

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

In [15]:

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

```
100%|██████████| 101/101 [00:00<00:00, 388290.29it/s]
```

In [ ]:

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

In [16]:

```
num_kps_akaze  = []



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

```
100%|██████████| 101/101 [00:00<00:00, 231996.00it/s]
```

In [15]:

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

```
100%|██████████| 101/101 [00:00<00:00, 402302.66it/s]
```

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

```
    num_kps_star.append(len(j))
```

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [16]:

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

```
100%|██████████| 101/101 [00:00<00:00, 422779.15it/s]
```

In [16]:

```
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 [17]:

```
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 [18]:

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

In [ ]:

```
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)
```

In [ ]:

```
H_left_orb = []
H_right_orb = []

num_matches_orb = []
num_good_matches_orb = []

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

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

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

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

In [20]:

```
H_left_akaze = []
H_right_akaze = []

num_matches_akaze = []
num_good_matches_akaze = []

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

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

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

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

```
    num_matches_akaze.append(matches)
    num_good_matches_akaze.append(gd_matches)
```

```
  2%|▏         | 1/51 [00:01<00:53,  1.06s/it]
```

```
Number of matches 20465
Number of matches After Lowe's Ratio 2247
Number of Robust matches 1631
```

```
  4%|▍         | 2/51 [00:02<01:00,  1.24s/it]
```

```
Number of matches 19280
Number of matches After Lowe's Ratio 2600
Number of Robust matches 1667
```

```
  6%|▌         | 3/51 [00:03<01:00,  1.27s/it]
```

```
Number of matches 21349
Number of matches After Lowe's Ratio 2768
Number of Robust matches 1957
```

```
  8%|▊         | 4/51 [00:05<01:06,  1.42s/it]
```

```
Number of matches 21345
Number of matches After Lowe's Ratio 4124
Number of Robust matches 3132
```

```
 10%|▉         | 5/51 [00:06<01:06,  1.45s/it]
```

```
Number of matches 20541
Number of matches After Lowe's Ratio 3174
Number of Robust matches 2271
```

```
 12%|█▏        | 6/51 [00:08<01:06,  1.47s/it]
```

```
Number of matches 19544
Number of matches After Lowe's Ratio 3406
Number of Robust matches 2447
```

```
 14%|█▎        | 7/51 [00:09<01:04,  1.46s/it]
```

```
Number of matches 19557
Number of matches After Lowe's Ratio 3165
Number of Robust matches 2512
```

```
 16%|█▌        | 8/51 [00:11<01:01,  1.42s/it]
```

```
Number of matches 19398
Number of matches After Lowe's Ratio 4086
Number of Robust matches 3101
```

```
 18%|█▊        | 9/51 [00:12<01:00,  1.43s/it]
```

```
Number of matches 19838
Number of matches After Lowe's Ratio 4548
Number of Robust matches 3222
```

```
 20%|█▉        | 10/51 [00:13<00:57,  1.41s/it]
```

Number of matches 19744
Number of matches After Lowe's Ratio 4021
Number of Robust matches 3321

22%|██ | 11/51 [00:15<00:55, 1.40s/it]

Number of matches 20624
Number of matches After Lowe's Ratio 3423
Number of Robust matches 2348

24%|██ | 12/51 [00:16<00:54, 1.41s/it]

Number of matches 19950
Number of matches After Lowe's Ratio 3061
Number of Robust matches 2221

25%|██ | 13/51 [00:18<00:54, 1.44s/it]

Number of matches 20566
Number of matches After Lowe's Ratio 2810
Number of Robust matches 1688

27%|██ | 14/51 [00:20<00:56, 1.53s/it]

Number of matches 20559
Number of matches After Lowe's Ratio 2376
Number of Robust matches 1599

29%|██ | 15/51 [00:21<00:54, 1.50s/it]

Number of matches 24258
Number of matches After Lowe's Ratio 2050
Number of Robust matches 1134

31%|███ | 16/51 [00:23<00:54, 1.57s/it]

Number of matches 20958
Number of matches After Lowe's Ratio 1793
Number of Robust matches 754

33%|███ | 17/51 [00:25<00:58, 1.71s/it]

Number of matches 22246
Number of matches After Lowe's Ratio 2174
Number of Robust matches 1102

35%|███ | 18/51 [00:26<00:55, 1.69s/it]

Number of matches 20947
Number of matches After Lowe's Ratio 2166
Number of Robust matches 943

37%|███ | 19/51 [00:28<00:52, 1.64s/it]

Number of matches 24081
Number of matches After Lowe's Ratio 1522
Number of Robust matches 607

Number of Robust matches 667

39%|███▉ | 20/51 [00:30<00:52, 1.68s/it]

Number of matches 22618
Number of matches After Lowe's Ratio 1557
Number of Robust matches 727

41%|████ | 21/51 [00:31<00:51, 1.72s/it]

Number of matches 23539
Number of matches After Lowe's Ratio 954
Number of Robust matches 328

43%|████▎ | 22/51 [00:33<00:50, 1.73s/it]

Number of matches 19832
Number of matches After Lowe's Ratio 476
Number of Robust matches 73

45%|████▌ | 23/51 [00:35<00:45, 1.62s/it]

Number of matches 19393
Number of matches After Lowe's Ratio 1889
Number of Robust matches 1071

47%|████▋ | 24/51 [00:36<00:41, 1.52s/it]

Number of matches 17976
Number of matches After Lowe's Ratio 2000
Number of Robust matches 1191

49%|████▉ | 25/51 [00:37<00:36, 1.42s/it]

Number of matches 19408
Number of matches After Lowe's Ratio 2052
Number of Robust matches 1304

51%|█████ | 26/51 [00:39<00:35, 1.44s/it]

Number of matches 23039
Number of matches After Lowe's Ratio 2408
Number of Robust matches 1370

53%|█████▎ | 27/51 [00:40<00:36, 1.52s/it]

Number of matches 26557
Number of matches After Lowe's Ratio 2007
Number of Robust matches 908

55%|█████▍ | 28/51 [00:43<00:40, 1.74s/it]

Number of matches 28674
Number of matches After Lowe's Ratio 2370
Number of Robust matches 800

57%|█████▋ | 29/51 [00:45<00:40, 1.85s/it]

57%|[redacted] | 29/51 [00:45<00:40, 1.85s/it]

```
Number of matches 25251
Number of matches After Lowe's Ratio 2276
Number of Robust matches 1089
```

59%|[redacted] | 30/51 [00:47<00:39, 1.88s/it]

```
Number of matches 22062
Number of matches After Lowe's Ratio 2516
Number of Robust matches 1233
```

61%|[redacted] | 31/51 [00:48<00:35, 1.79s/it]

```
Number of matches 20521
Number of matches After Lowe's Ratio 2763
Number of Robust matches 1726
```

63%|[redacted] | 32/51 [00:50<00:32, 1.69s/it]

```
Number of matches 19126
Number of matches After Lowe's Ratio 2912
Number of Robust matches 1836
```

65%|[redacted] | 33/51 [00:51<00:28, 1.58s/it]

```
Number of matches 20186
Number of matches After Lowe's Ratio 3556
Number of Robust matches 2680
```

67%|[redacted] | 34/51 [00:53<00:28, 1.69s/it]

```
Number of matches 21213
Number of matches After Lowe's Ratio 3206
Number of Robust matches 2033
```

69%|[redacted] | 35/51 [00:54<00:26, 1.64s/it]

```
Number of matches 21932
Number of matches After Lowe's Ratio 3698
Number of Robust matches 2462
```

71%|[redacted] | 36/51 [00:57<00:27, 1.82s/it]

```
Number of matches 21264
Number of matches After Lowe's Ratio 3861
Number of Robust matches 2570
```

73%|[redacted] | 37/51 [00:58<00:24, 1.72s/it]

```
Number of matches 20833
Number of matches After Lowe's Ratio 3430
Number of Robust matches 2329
```

75%|[redacted] | 38/51 [01:00<00:21, 1.63s/it]

```
Number of matches 18699
Number of matches After Lowe's Ratio 2091
```

```
Number of Robust matches 1320

 76%|███████▌  | 39/51 [01:01<00:18,  1.57s/it]

Number of matches 17733
Number of matches After Lowe's Ratio 3728
Number of Robust matches 2775

 78%|███████▊  | 40/51 [01:02<00:16,  1.47s/it]

Number of matches 18293
Number of matches After Lowe's Ratio 3590
Number of Robust matches 3077

 80%|████████  | 41/51 [01:04<00:14,  1.48s/it]

Number of matches 16473
Number of matches After Lowe's Ratio 2100
Number of Robust matches 1614

 82%|████████▏ | 42/51 [01:05<00:12,  1.38s/it]

Number of matches 17759
Number of matches After Lowe's Ratio 2215
Number of Robust matches 1616

 84%|████████▍ | 43/51 [01:06<00:10,  1.33s/it]

Number of matches 18253
Number of matches After Lowe's Ratio 3005
Number of Robust matches 2313

 86%|████████▋ | 44/51 [01:08<00:09,  1.36s/it]

Number of matches 18717
Number of matches After Lowe's Ratio 1934
Number of Robust matches 1140

 88%|████████▊ | 45/51 [01:09<00:08,  1.35s/it]

Number of matches 18943
Number of matches After Lowe's Ratio 2377
Number of Robust matches 1691

 90%|█████████ | 46/51 [01:10<00:06,  1.33s/it]

Number of matches 18446
Number of matches After Lowe's Ratio 2212
Number of Robust matches 1328

 92%|█████████▏| 47/51 [01:11<00:05,  1.33s/it]

Number of matches 19154
Number of matches After Lowe's Ratio 2731
Number of Robust matches 1355
```

```
 94%|█████████ | 48/51 [01:13<00:03,  1.33s/it]

Number of matches 20674
Number of matches After Lowe's Ratio 1805
Number of Robust matches 779


 96%|█████████ | 49/51 [01:15<00:02,  1.49s/it]

Number of matches 20317
Number of matches After Lowe's Ratio 2136
Number of Robust matches 925


 98%|█████████ | 50/51 [01:16<00:01,  1.53s/it]
  0%|          | 0/50 [00:00<?, ?it/s]

Number of matches 17535
Number of matches After Lowe's Ratio 738
Number of Robust matches 196


  2%|▏         | 1/50 [00:01<00:53,  1.10s/it]

Number of matches 21469
Number of matches After Lowe's Ratio 1232
Number of Robust matches 704


  4%|▍         | 2/50 [00:02<01:01,  1.29s/it]

Number of matches 15748
Number of matches After Lowe's Ratio 1871
Number of Robust matches 1189


  6%|▌         | 3/50 [00:03<00:55,  1.18s/it]

Number of matches 20428
Number of matches After Lowe's Ratio 1277
Number of Robust matches 654


  8%|▊         | 4/50 [00:05<01:00,  1.32s/it]

Number of matches 19049
Number of matches After Lowe's Ratio 2520
Number of Robust matches 1507


 10%|▉         | 5/50 [00:06<00:58,  1.30s/it]

Number of matches 18754
Number of matches After Lowe's Ratio 2341
Number of Robust matches 1155


 12%|█▏        | 6/50 [00:07<00:56,  1.28s/it]

Number of matches 18110
Number of matches After Lowe's Ratio 2267
Number of Robust matches 1154


 14%|█▍        | 7/50 [00:09<00:57,  1.34s/it]

Number of matches 18547
```

```
Number of matches After Lowe's Ratio 2210
Number of Robust matches 1399
```

```
 16%|▉         | 8/50 [00:10<00:57,  1.37s/it]
```

```
Number of matches 21720
Number of matches After Lowe's Ratio 493
Number of Robust matches 34
```

```
 18%|▉         | 9/50 [00:12<01:03,  1.56s/it]
```

```
Number of matches 16167
Number of matches After Lowe's Ratio 826
Number of Robust matches 282
```

```
 20%|█         | 10/50 [00:13<00:56,  1.41s/it]
```

```
Number of matches 16819
Number of matches After Lowe's Ratio 948
Number of Robust matches 421
```

```
 22%|██        | 11/50 [00:14<00:51,  1.33s/it]
```

```
Number of matches 18233
Number of matches After Lowe's Ratio 1422
Number of Robust matches 824
```

```
 24%|██        | 12/50 [00:15<00:49,  1.30s/it]
```

```
Number of matches 20771
Number of matches After Lowe's Ratio 2613
Number of Robust matches 1736
```

```
 26%|██        | 13/50 [00:17<00:50,  1.36s/it]
```

```
Number of matches 18297
Number of matches After Lowe's Ratio 2757
Number of Robust matches 2066
```

```
 28%|██        | 14/50 [00:18<00:47,  1.32s/it]
```

```
Number of matches 15660
Number of matches After Lowe's Ratio 1374
Number of Robust matches 863
```

```
 30%|███       | 15/50 [00:19<00:44,  1.28s/it]
```

```
Number of matches 19968
Number of matches After Lowe's Ratio 623
Number of Robust matches 218
```

```
 32%|███       | 16/50 [00:21<00:44,  1.31s/it]
```

```
Number of matches 15728
Number of matches After Lowe's Ratio 2079
Number of Robust matches 1414
```

34%|███ | 17/50 [00:22<00:40, 1.24s/it]

```
Number of matches 21692
Number of matches After Lowe's Ratio 1834
Number of Robust matches 1234
```

36%|███ | 18/50 [00:23<00:42, 1.33s/it]

```
Number of matches 21865
Number of matches After Lowe's Ratio 3695
Number of Robust matches 2800
```

38%|███ | 19/50 [00:25<00:44, 1.43s/it]

```
Number of matches 21455
Number of matches After Lowe's Ratio 5039
Number of Robust matches 3608
```

40%|████ | 20/50 [00:27<00:43, 1.45s/it]

```
Number of matches 19571
Number of matches After Lowe's Ratio 4118
Number of Robust matches 3154
```

42%|████ | 21/50 [00:28<00:41, 1.42s/it]

```
Number of matches 20134
Number of matches After Lowe's Ratio 3983
Number of Robust matches 2691
```

44%|████ | 22/50 [00:29<00:40, 1.45s/it]

```
Number of matches 21099
Number of matches After Lowe's Ratio 2882
Number of Robust matches 2074
```

46%|████ | 23/50 [00:31<00:41, 1.55s/it]

```
Number of matches 22136
Number of matches After Lowe's Ratio 3600
Number of Robust matches 2639
```

48%|████ | 24/50 [00:33<00:40, 1.56s/it]

```
Number of matches 23198
Number of matches After Lowe's Ratio 2987
Number of Robust matches 2021
```

50%|█████ | 25/50 [00:34<00:40, 1.61s/it]

```
Number of matches 24310
Number of matches After Lowe's Ratio 2984
Number of Robust matches 1573
```

52%|█████ | 26/50 [00:36<00:41, 1.72s/it]

```
Number of matches 24654
```

Number of matches 24654
Number of matches After Lowe's Ratio 3148
Number of Robust matches 1698

54%|[redacted] | 27/50 [00:38<00:39, 1.74s/it]

Number of matches 22984
Number of matches After Lowe's Ratio 3152
Number of Robust matches 1662

56%|[redacted] | 28/50 [00:40<00:37, 1.72s/it]

Number of matches 21716
Number of matches After Lowe's Ratio 2724
Number of Robust matches 1311

58%|[redacted] | 29/50 [00:42<00:37, 1.77s/it]

Number of matches 22411
Number of matches After Lowe's Ratio 3286
Number of Robust matches 1623

60%|[redacted] | 30/50 [00:44<00:38, 1.93s/it]

Number of matches 20504
Number of matches After Lowe's Ratio 2743
Number of Robust matches 1169

62%|[redacted] | 31/50 [00:46<00:33, 1.78s/it]

Number of matches 20204
Number of matches After Lowe's Ratio 2642
Number of Robust matches 1030

64%|[redacted] | 32/50 [00:47<00:30, 1.68s/it]

Number of matches 20233
Number of matches After Lowe's Ratio 2172
Number of Robust matches 1156

66%|[redacted] | 33/50 [00:49<00:27, 1.63s/it]

Number of matches 27287
Number of matches After Lowe's Ratio 797
Number of Robust matches 203

68%|[redacted] | 34/50 [00:51<00:29, 1.81s/it]

Number of matches 25609
Number of matches After Lowe's Ratio 1247
Number of Robust matches 385

70%|[redacted] | 35/50 [00:53<00:29, 1.97s/it]

Number of matches 29338
Number of matches After Lowe's Ratio 481
Number of Robust matches 5

72%|███████ | 36/50 [00:55<00:29, 2.08s/it]

```
Number of matches 24561
Number of matches After Lowe's Ratio 1198
Number of Robust matches 397
```

74%|███████ | 37/50 [00:57<00:26, 2.01s/it]

```
Number of matches 22164
Number of matches After Lowe's Ratio 2260
Number of Robust matches 1073
```

76%|███████ | 38/50 [00:59<00:23, 1.93s/it]

```
Number of matches 20348
Number of matches After Lowe's Ratio 2259
Number of Robust matches 973
```

78%|███████ | 39/50 [01:00<00:19, 1.78s/it]

```
Number of matches 19519
Number of matches After Lowe's Ratio 1972
Number of Robust matches 849
```

80%|████████ | 40/50 [01:02<00:16, 1.66s/it]

```
Number of matches 20911
Number of matches After Lowe's Ratio 2024
Number of Robust matches 729
```

82%|████████ | 41/50 [01:03<00:15, 1.67s/it]

```
Number of matches 20565
Number of matches After Lowe's Ratio 1760
Number of Robust matches 698
```

84%|████████ | 42/50 [01:05<00:13, 1.66s/it]

```
Number of matches 20347
Number of matches After Lowe's Ratio 3746
Number of Robust matches 1570
```

86%|████████ | 43/50 [01:07<00:11, 1.61s/it]

```
Number of matches 22441
Number of matches After Lowe's Ratio 1810
Number of Robust matches 818
```

88%|████████ | 44/50 [01:08<00:09, 1.61s/it]

```
Number of matches 21870
Number of matches After Lowe's Ratio 3146
Number of Robust matches 1725
```

90%|████████ | 45/50 [01:10<00:07, 1.58s/it]

```
Number of matches 20671
Number of matches After Lowe's Ratio 2296
Number of Robust matches 1293
```

```
 92%|█████████ | 46/50 [01:11<00:06,  1.53s/it]
```

```
Number of matches 18732
Number of matches After Lowe's Ratio 2371
Number of Robust matches 1514
```

```
 94%|█████████ | 47/50 [01:13<00:04,  1.50s/it]
```

```
Number of matches 18770
Number of matches After Lowe's Ratio 1822
Number of Robust matches 1323
```

```
 96%|█████████ | 48/50 [01:14<00:02,  1.44s/it]
```

```
Number of matches 18471
Number of matches After Lowe's Ratio 1874
Number of Robust matches 1217
```

```
 98%|█████████ | 49/50 [01:15<00:01,  1.55s/it]
```

```
Number of matches 17991
Number of matches After Lowe's Ratio 2162
Number of Robust matches 1552
```

In [ ]:

```
H_left_kaze = []
H_right_kaze = []

num_matches_kaze = []
num_good_matches_kaze = []

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

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

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

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

In [19]:

```
H_left_freak = []
H_right_freak = []
```

```
num_matches_freak = []
num_good_matches_freak = []

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

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

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

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

```
  2%|          | 1/51 [00:02<01:51,  2.23s/it]
Number of matches 30963
Number of matches After Lowe's Ratio 1989
Number of Robust matches 1063

  4%|          | 2/51 [00:04<01:59,  2.45s/it]
Number of matches 30334
Number of matches After Lowe's Ratio 1980
Number of Robust matches 1153

  6%|          | 3/51 [00:08<02:14,  2.80s/it]
Number of matches 35225
Number of matches After Lowe's Ratio 2066
Number of Robust matches 956

  8%|          | 4/51 [00:11<02:25,  3.09s/it]
Number of matches 34588
Number of matches After Lowe's Ratio 2630
Number of Robust matches 1743

 10%|          | 5/51 [00:14<02:20,  3.04s/it]
Number of matches 32541
Number of matches After Lowe's Ratio 2201
Number of Robust matches 1205

 12%|          | 6/51 [00:17<02:11,  2.92s/it]
Number of matches 28721
Number of matches After Lowe's Ratio 2114
Number of Robust matches 1159

 14%|          | 7/51 [00:19<02:02,  2.79s/it]
```

```
Number of matches 30225
Number of matches After Lowe's Ratio 2198
Number of Robust matches 1272
```

```
 16%|█▌        | 8/51 [00:22<02:00,  2.80s/it]
```

```
Number of matches 30991
Number of matches After Lowe's Ratio 2537
Number of Robust matches 1528
```

```
 18%|█▊        | 9/51 [00:25<01:54,  2.72s/it]
```

```
Number of matches 29265
Number of matches After Lowe's Ratio 2787
Number of Robust matches 1671
```

```
 20%|██        | 10/51 [00:27<01:49,  2.66s/it]
```

```
Number of matches 29356
Number of matches After Lowe's Ratio 2183
Number of Robust matches 1273
```

```
 22%|██▏       | 11/51 [00:30<01:44,  2.60s/it]
```

```
Number of matches 30010
Number of matches After Lowe's Ratio 1764
Number of Robust matches 821
```

```
 24%|██▍       | 12/51 [00:32<01:44,  2.67s/it]
```

```
Number of matches 29570
Number of matches After Lowe's Ratio 1823
Number of Robust matches 828
```

```
 25%|██▌       | 13/51 [00:35<01:41,  2.68s/it]
```

```
Number of matches 30831
Number of matches After Lowe's Ratio 2044
Number of Robust matches 1017
```

```
 27%|██▋       | 14/51 [00:38<01:43,  2.81s/it]
```

```
Number of matches 33305
Number of matches After Lowe's Ratio 1758
Number of Robust matches 811
```

```
 29%|██▉       | 15/51 [00:41<01:45,  2.93s/it]
```

```
Number of matches 41880
Number of matches After Lowe's Ratio 1803
Number of Robust matches 608
```

```
 31%|███       | 16/51 [00:45<01:53,  3.24s/it]
```

```
Number of matches 35904
Number of matches After Lowe's Ratio 1755
Number of Robust matches 548
```

Number of Robust matches 548

33%|███ | 17/51 [00:48<01:47, 3.18s/it]

Number of matches 30389
Number of matches After Lowe's Ratio 1716
Number of Robust matches 688

35%|███ | 18/51 [00:51<01:39, 3.02s/it]

Number of matches 28704
Number of matches After Lowe's Ratio 1403
Number of Robust matches 359

37%|███ | 19/51 [00:54<01:32, 2.90s/it]

Number of matches 31542
Number of matches After Lowe's Ratio 991
Number of Robust matches 252

39%|███ | 20/51 [00:56<01:27, 2.83s/it]

Number of matches 31853
Number of matches After Lowe's Ratio 1369
Number of Robust matches 433

41%|███ | 21/51 [00:59<01:24, 2.83s/it]

Number of matches 32825
Number of matches After Lowe's Ratio 866
Number of Robust matches 126

43%|███ | 22/51 [01:02<01:20, 2.77s/it]

Number of matches 22605
Number of matches After Lowe's Ratio 512
Number of Robust matches 28

45%|███ | 23/51 [01:04<01:09, 2.47s/it]

Number of matches 23139
Number of matches After Lowe's Ratio 1334
Number of Robust matches 431

47%|███ | 24/51 [01:06<01:02, 2.31s/it]

Number of matches 19638
Number of matches After Lowe's Ratio 1239
Number of Robust matches 502

49%|███ | 25/51 [01:07<00:54, 2.11s/it]

Number of matches 24280
Number of matches After Lowe's Ratio 1330
Number of Robust matches 473

51%|[redacted]| 26/51 [01:09<00:54, 2.16s/it]

```
Number of matches 29286
Number of matches After Lowe's Ratio 1620
Number of Robust matches 547
```

53%|[redacted]| 27/51 [01:12<00:56, 2.36s/it]

```
Number of matches 41948
Number of matches After Lowe's Ratio 1624
Number of Robust matches 443
```

55%|[redacted]| 28/51 [01:17<01:08, 3.00s/it]

```
Number of matches 45777
Number of matches After Lowe's Ratio 2082
Number of Robust matches 563
```

57%|[redacted]| 29/51 [01:21<01:14, 3.38s/it]

```
Number of matches 41498
Number of matches After Lowe's Ratio 1897
Number of Robust matches 539
```

59%|[redacted]| 30/51 [01:25<01:13, 3.49s/it]

```
Number of matches 32765
Number of matches After Lowe's Ratio 1963
Number of Robust matches 683
```

61%|[redacted]| 31/51 [01:28<01:07, 3.35s/it]

```
Number of matches 30811
Number of matches After Lowe's Ratio 2127
Number of Robust matches 1067
```

63%|[redacted]| 32/51 [01:30<00:58, 3.10s/it]

```
Number of matches 28921
Number of matches After Lowe's Ratio 2084
Number of Robust matches 967
```

65%|[redacted]| 33/51 [01:33<00:52, 2.92s/it]

```
Number of matches 28541
Number of matches After Lowe's Ratio 2300
Number of Robust matches 1337
```

67%|[redacted]| 34/51 [01:35<00:47, 2.78s/it]

```
Number of matches 34315
Number of matches After Lowe's Ratio 2273
Number of Robust matches 1125
```

69%|[redacted]| 35/51 [01:39<00:46, 2.93s/it]

```
Number of matches 37159
Number of matches After Lowe's Ratio 2701
```

```
Number of Robust matches 1154

 71%|███████   | 36/51 [01:43<00:48,  3.26s/it]

Number of matches 35439
Number of matches After Lowe's Ratio 2474
Number of Robust matches 1151

 73%|███████   | 37/51 [01:46<00:44,  3.21s/it]

Number of matches 36939
Number of matches After Lowe's Ratio 2565
Number of Robust matches 1078

 75%|███████   | 38/51 [01:49<00:43,  3.31s/it]

Number of matches 30760
Number of matches After Lowe's Ratio 1669
Number of Robust matches 747

 76%|███████   | 39/51 [01:52<00:37,  3.09s/it]

Number of matches 29514
Number of matches After Lowe's Ratio 2856
Number of Robust matches 1773

 78%|███████   | 40/51 [01:54<00:31,  2.87s/it]

Number of matches 28402
Number of matches After Lowe's Ratio 2520
Number of Robust matches 1533

 80%|████████  | 41/51 [01:56<00:26,  2.68s/it]

Number of matches 25855
Number of matches After Lowe's Ratio 1488
Number of Robust matches 705

 82%|████████  | 42/51 [01:59<00:22,  2.53s/it]

Number of matches 25442
Number of matches After Lowe's Ratio 1541
Number of Robust matches 761

 84%|████████  | 43/51 [02:01<00:19,  2.44s/it]

Number of matches 25540
Number of matches After Lowe's Ratio 1967
Number of Robust matches 1047

 86%|████████  | 44/51 [02:03<00:16,  2.34s/it]

Number of matches 31109
Number of matches After Lowe's Ratio 1914
Number of Robust matches 845
```

```
 88%|▉| 45/51 [02:05<00:14,  2.39s/it]

Number of matches 25420
Number of matches After Lowe's Ratio 1734
Number of Robust matches 1047


 90%|▉| 46/51 [02:08<00:11,  2.34s/it]

Number of matches 27131
Number of matches After Lowe's Ratio 1672
Number of Robust matches 570


 92%|▉| 47/51 [02:10<00:09,  2.36s/it]

Number of matches 30777
Number of matches After Lowe's Ratio 2582
Number of Robust matches 950


 94%|▉| 48/51 [02:13<00:07,  2.47s/it]

Number of matches 31215
Number of matches After Lowe's Ratio 1673
Number of Robust matches 530


 96%|▉| 49/51 [02:16<00:05,  2.73s/it]

Number of matches 34016
Number of matches After Lowe's Ratio 2530
Number of Robust matches 749


 98%|▉| 50/51 [02:19<00:02,  2.79s/it]
  0%|          | 0/50 [00:00<?, ?it/s]

Number of matches 23270
Number of matches After Lowe's Ratio 866
Number of Robust matches 138


  2%|▏         | 1/50 [00:02<01:38,  2.00s/it]

Number of matches 28885
Number of matches After Lowe's Ratio 1186
Number of Robust matches 521


  4%|▍         | 2/50 [00:04<01:54,  2.38s/it]

Number of matches 20799
Number of matches After Lowe's Ratio 1332
Number of Robust matches 692


  6%|▌         | 3/50 [00:06<01:37,  2.07s/it]

Number of matches 29762
Number of matches After Lowe's Ratio 1188
Number of Robust matches 376


  8%|▊         | 4/50 [00:08<01:41,  2.20s/it]

Number of matches 24534
```

Number of matches After Lowe's Ratio 1507
Number of Robust matches 709

10%|▉ | 5/50 [00:10<01:34, 2.11s/it]

Number of matches 26179
Number of matches After Lowe's Ratio 1624
Number of Robust matches 524

12%|▉ | 6/50 [00:12<01:33, 2.13s/it]

Number of matches 21791
Number of matches After Lowe's Ratio 1331
Number of Robust matches 501

14%|▉ | 7/50 [00:14<01:29, 2.07s/it]

Number of matches 23985
Number of matches After Lowe's Ratio 1338
Number of Robust matches 579

16%|▉ | 8/50 [00:16<01:25, 2.03s/it]

Number of matches 29091
Number of matches After Lowe's Ratio 681
Number of Robust matches 8

18%|▉ | 9/50 [00:19<01:26, 2.11s/it]

Number of matches 23038
Number of matches After Lowe's Ratio 816
Number of Robust matches 185

20%|▉ | 10/50 [00:20<01:21, 2.05s/it]

Number of matches 21727
Number of matches After Lowe's Ratio 669
Number of Robust matches 167

22%|▉ | 11/50 [00:22<01:15, 1.94s/it]

Number of matches 23257
Number of matches After Lowe's Ratio 889
Number of Robust matches 253

24%|▉ | 12/50 [00:24<01:16, 2.02s/it]

Number of matches 33002
Number of matches After Lowe's Ratio 1687
Number of Robust matches 830

26%|▉ | 13/50 [00:28<01:30, 2.45s/it]

Number of matches 26873
Number of matches After Lowe's Ratio 1814
Number of Robust matches 1045

28%|██▊ | 14/50 [00:30<01:24, 2.33s/it]

```
Number of matches 24439
Number of matches After Lowe's Ratio 1101
Number of Robust matches 527
```

30%|███ | 15/50 [00:32<01:16, 2.20s/it]

```
Number of matches 22762
Number of matches After Lowe's Ratio 632
Number of Robust matches 156
```

32%|███▏ | 16/50 [00:33<01:09, 2.04s/it]

```
Number of matches 19743
Number of matches After Lowe's Ratio 1460
Number of Robust matches 909
```

34%|███▍ | 17/50 [00:35<01:06, 2.03s/it]

```
Number of matches 29263
Number of matches After Lowe's Ratio 1152
Number of Robust matches 445
```

36%|███▌ | 18/50 [00:38<01:08, 2.15s/it]

```
Number of matches 29662
Number of matches After Lowe's Ratio 2355
Number of Robust matches 1398
```

38%|███▊ | 19/50 [00:40<01:09, 2.26s/it]

```
Number of matches 31340
Number of matches After Lowe's Ratio 2436
Number of Robust matches 1743
```

40%|████ | 20/50 [00:43<01:10, 2.35s/it]

```
Number of matches 28069
Number of matches After Lowe's Ratio 1984
Number of Robust matches 1200
```

42%|████▏ | 21/50 [00:45<01:09, 2.40s/it]

```
Number of matches 30957
Number of matches After Lowe's Ratio 2200
Number of Robust matches 1536
```

44%|████▍ | 22/50 [00:48<01:11, 2.56s/it]

```
Number of matches 30565
Number of matches After Lowe's Ratio 1679
Number of Robust matches 882
```

46%|████▌ | 23/50 [00:51<01:09, 2.58s/it]

```
Number of matches 32974
Number of matches After Lowe's Ratio 2125
Number of Robust matches 1229


 48%|████▊     | 24/50 [00:54<01:11,  2.73s/it]

Number of matches 36549
Number of matches After Lowe's Ratio 2051
Number of Robust matches 1008


 50%|█████     | 25/50 [00:58<01:14,  2.99s/it]

Number of matches 35159
Number of matches After Lowe's Ratio 2170
Number of Robust matches 870


 52%|█████▏    | 26/50 [01:02<01:17,  3.24s/it]

Number of matches 38169
Number of matches After Lowe's Ratio 2397
Number of Robust matches 1054


 54%|█████▍    | 27/50 [01:05<01:15,  3.30s/it]

Number of matches 33493
Number of matches After Lowe's Ratio 2151
Number of Robust matches 872


 56%|█████▌    | 28/50 [01:08<01:10,  3.21s/it]

Number of matches 27254
Number of matches After Lowe's Ratio 1779
Number of Robust matches 673


 58%|█████▊    | 29/50 [01:10<01:02,  2.97s/it]

Number of matches 29921
Number of matches After Lowe's Ratio 2036
Number of Robust matches 679


 60%|██████    | 30/50 [01:13<00:56,  2.82s/it]

Number of matches 27914
Number of matches After Lowe's Ratio 1944
Number of Robust matches 628


 62%|██████▏   | 31/50 [01:15<00:50,  2.63s/it]

Number of matches 22382
Number of matches After Lowe's Ratio 1393
Number of Robust matches 454


 64%|██████▍   | 32/50 [01:17<00:42,  2.37s/it]

Number of matches 26806
Number of matches After Lowe's Ratio 1380
Number of Robust matches 456
```

66%|██████ | 33/50 [01:20<00:42, 2.51s/it]

```
Number of matches 43407
Number of matches After Lowe's Ratio 886
Number of Robust matches 53
```

68%|██████ | 34/50 [01:24<00:47, 3.00s/it]

```
Number of matches 39639
Number of matches After Lowe's Ratio 1032
Number of Robust matches 258
```

70%|███████ | 35/50 [01:28<00:49, 3.31s/it]

```
Number of matches 44415
Number of matches After Lowe's Ratio 733
Number of Robust matches 7
```

72%|███████ | 36/50 [01:33<00:53, 3.79s/it]

```
Number of matches 34575
Number of matches After Lowe's Ratio 978
Number of Robust matches 183
```

74%|███████ | 37/50 [01:36<00:46, 3.60s/it]

```
Number of matches 32318
Number of matches After Lowe's Ratio 1672
Number of Robust matches 485
```

76%|███████ | 38/50 [01:39<00:40, 3.35s/it]

```
Number of matches 30468
Number of matches After Lowe's Ratio 1794
Number of Robust matches 461
```

78%|███████ | 39/50 [01:41<00:34, 3.16s/it]

```
Number of matches 26838
Number of matches After Lowe's Ratio 1530
Number of Robust matches 423
```

80%|████████ | 40/50 [01:44<00:28, 2.85s/it]

```
Number of matches 24184
Number of matches After Lowe's Ratio 1216
Number of Robust matches 294
```

82%|████████ | 41/50 [01:46<00:23, 2.62s/it]

```
Number of matches 25941
Number of matches After Lowe's Ratio 1191
Number of Robust matches 316
```

84%|████████ | 42/50 [01:48<00:19, 2.46s/it]

```
Number of matches 27985
Number of matches After Lowe's Ratio 2088
Number of Robust matches 622
```

```
 86%|████████  | 43/50 [01:50<00:17,  2.44s/it]
```

```
Number of matches 29906
Number of matches After Lowe's Ratio 1419
Number of Robust matches 361
```

```
 88%|████████▊ | 44/50 [01:53<00:15,  2.53s/it]
```

```
Number of matches 26647
Number of matches After Lowe's Ratio 1659
Number of Robust matches 616
```

```
 90%|█████████ | 45/50 [01:55<00:12,  2.49s/it]
```

```
Number of matches 30823
Number of matches After Lowe's Ratio 1837
Number of Robust matches 608
```

```
 92%|█████████▏| 46/50 [01:58<00:09,  2.48s/it]
```

```
Number of matches 23861
Number of matches After Lowe's Ratio 1340
Number of Robust matches 529
```

```
 94%|█████████▍| 47/50 [02:00<00:06,  2.30s/it]
```

```
Number of matches 26390
Number of matches After Lowe's Ratio 1064
Number of Robust matches 500
```

```
 96%|█████████▌| 48/50 [02:02<00:04,  2.30s/it]
```

```
Number of matches 26937
Number of matches After Lowe's Ratio 1336
Number of Robust matches 733
```

```
 98%|█████████▊| 49/50 [02:05<00:02,  2.56s/it]
```

```
Number of matches 26134
Number of matches After Lowe's Ratio 1284
Number of Robust matches 585
```

In [ ]:

```
H_left_mser = []
H_right_mser = []

num_matches_mser = []
num_good_matches_mser = []

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

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

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

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

In [23]:

```
H_left_superpoint = []
H_right_superpoint = []

num_matches_superpoint = []
num_good_matches_superpoint = []

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

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

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

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

```
  2%|          | 1/51 [00:00<00:22,  2.25it/s]
```

```
Number of matches 2000
Number of matches After Lowe's Ratio 279
Number of Robust matches 62
```

```
  4%|          | 2/51 [00:00<00:21,  2.26it/s]
```

```
Number of matches 2000
Number of matches After Lowe's Ratio 118
Number of Robust matches 38
```

```
  6%|          | 3/51 [00:01<00:21,  2.23it/s]
```

```
Number of matches 2000
Number of matches After Lowe's Ratio 126
Number of Robust matches 33
```

```
  8%|▊         | 4/51 [00:01<00:20,  2.25it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 223
Number of Robust matches 58

 10%|▉         | 5/51 [00:02<00:20,  2.24it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 212
Number of Robust matches 54

 12%|█         | 6/51 [00:02<00:20,  2.22it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 130
Number of Robust matches 34

 14%|█▏        | 7/51 [00:03<00:19,  2.22it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 230
Number of Robust matches 46

 16%|█▎        | 8/51 [00:03<00:20,  2.11it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 255
Number of Robust matches 51

 18%|█▍        | 9/51 [00:04<00:27,  1.55it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 325
Number of Robust matches 69

 20%|█▌        | 10/51 [00:05<00:25,  1.59it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 219
Number of Robust matches 51

 22%|█▋        | 11/51 [00:05<00:22,  1.74it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 200
Number of Robust matches 51

 24%|█▊        | 12/51 [00:06<00:20,  1.86it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 236
Number of Robust matches 42

 25%|█▉        | 13/51 [00:06<00:19,  1.95it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 115
```

```
Number of Robust matches 24


 27%|██▋       | 14/51 [00:07<00:18,  2.02it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 168
Number of Robust matches 26


 29%|██▉       | 15/51 [00:07<00:17,  2.09it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 63
Number of Robust matches 18


 31%|███       | 16/51 [00:07<00:16,  2.11it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 69
Number of Robust matches 22


 33%|███▎      | 17/51 [00:08<00:15,  2.13it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 209
Number of Robust matches 50


 35%|███▌      | 18/51 [00:08<00:15,  2.12it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 84
Number of Robust matches 19


 37%|███▋      | 19/51 [00:09<00:15,  2.13it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 47
Number of Robust matches 7


 39%|███▉      | 20/51 [00:09<00:14,  2.16it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 161
Number of Robust matches 24


 41%|████      | 21/51 [00:10<00:13,  2.15it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 91
Number of Robust matches 22


 43%|████▎     | 22/51 [00:10<00:13,  2.21it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 12
Number of Robust matches 4
```

45%|████ | 23/51 [00:11<00:12, 2.21it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 235
Number of Robust matches 40
```

47%|████ | 24/51 [00:11<00:12, 2.20it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 245
Number of Robust matches 36
```

49%|████ | 25/51 [00:12<00:11, 2.18it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 186
Number of Robust matches 30
```

51%|████ | 26/51 [00:12<00:11, 2.15it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 252
Number of Robust matches 49
```

53%|████ | 27/51 [00:13<00:11, 2.14it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 197
Number of Robust matches 23
```

55%|████ | 28/51 [00:13<00:10, 2.12it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 233
Number of Robust matches 28
```

57%|████ | 29/51 [00:14<00:10, 2.14it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 153
Number of Robust matches 30
```

59%|████ | 30/51 [00:14<00:09, 2.18it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 210
Number of Robust matches 41
```

61%|████ | 31/51 [00:14<00:09, 2.20it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 231
Number of Robust matches 50
```

63%|████ | 32/51 [00:15<00:10, 1.90it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 229
```

Number of Robust matches 57

65%|▋| 33/51 [00:16<00:09, 1.98it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 289
Number of Robust matches 69

67%|▋| 34/51 [00:16<00:08, 2.05it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 166
Number of Robust matches 29

69%|▋| 35/51 [00:16<00:07, 2.07it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 195
Number of Robust matches 33

71%|▋| 36/51 [00:17<00:07, 2.11it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 208
Number of Robust matches 45

73%|▋| 37/51 [00:17<00:06, 2.16it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 191
Number of Robust matches 43

75%|▋| 38/51 [00:18<00:06, 2.10it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 127
Number of Robust matches 24

76%|▋| 39/51 [00:18<00:05, 2.12it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 367
Number of Robust matches 77

78%|▋| 40/51 [00:19<00:05, 2.14it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 310
Number of Robust matches 55

80%|▊| 41/51 [00:19<00:04, 2.17it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 194
Number of Robust matches 35

```
 82%|████████  | 42/51 [00:20<00:04,  2.16it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 180
Number of Robust matches 39

 84%|████████  | 43/51 [00:20<00:03,  2.19it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 225
Number of Robust matches 43

 86%|████████  | 44/51 [00:21<00:03,  2.21it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 95
Number of Robust matches 27

 88%|████████  | 45/51 [00:21<00:02,  2.20it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 152
Number of Robust matches 33

 90%|█████████ | 46/51 [00:21<00:02,  2.20it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 149
Number of Robust matches 32

 92%|█████████ | 47/51 [00:22<00:01,  2.21it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 246
Number of Robust matches 60

 94%|█████████ | 48/51 [00:22<00:01,  2.22it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 85
Number of Robust matches 12

 96%|█████████ | 49/51 [00:23<00:00,  2.15it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 200
Number of Robust matches 30

 98%|█████████ | 50/51 [00:23<00:00,  2.10it/s]
  0%|          | 0/50 [00:00<?, ?it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 30
Number of Robust matches 7

  2%|          | 1/50 [00:00<00:22,  2.16it/s]
```

```
Number of matches 2000
Number of matches After Lowe's Ratio 73
Number of Robust matches 20


  4%|▍         | 2/50 [00:00<00:22,  2.09it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 81
Number of Robust matches 17


  6%|▌         | 3/50 [00:01<00:22,  2.06it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 92
Number of Robust matches 21


  8%|▊         | 4/50 [00:01<00:22,  2.08it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 140
Number of Robust matches 33


 10%|▉         | 5/50 [00:02<00:25,  1.79it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 236
Number of Robust matches 46


 12%|█▏        | 6/50 [00:03<00:23,  1.84it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 141
Number of Robust matches 33


 14%|█▍        | 7/50 [00:03<00:22,  1.93it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 205
Number of Robust matches 45


 16%|█▌        | 8/50 [00:04<00:21,  1.99it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 14
Number of Robust matches 4


 18%|█▊        | 9/50 [00:04<00:20,  1.99it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 58
Number of Robust matches 10


 20%|██        | 10/50 [00:05<00:19,  2.05it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 49
Number of Robust matches 13
```

```
 22%|██        | 11/50 [00:05<00:19,  2.02it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 212
Number of Robust matches 33

 24%|██        | 12/50 [00:06<00:18,  2.04it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 281
Number of Robust matches 73

 26%|██        | 13/50 [00:06<00:18,  2.05it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 329
Number of Robust matches 78

 28%|██        | 14/50 [00:06<00:17,  2.05it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 115
Number of Robust matches 24

 30%|███       | 15/50 [00:07<00:16,  2.09it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 30
Number of Robust matches 8

 32%|███       | 16/50 [00:07<00:16,  2.10it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 216
Number of Robust matches 54

 34%|███       | 17/50 [00:08<00:15,  2.13it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 117
Number of Robust matches 26

 36%|███       | 18/50 [00:08<00:14,  2.16it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 387
Number of Robust matches 85

 38%|███       | 19/50 [00:09<00:14,  2.19it/s]
Number of matches 2000
Number of matches After Lowe's Ratio 256
Number of Robust matches 65

 40%|████      | 20/50 [00:09<00:13,  2.16it/s]
```

```
Number of matches 2000
Number of matches After Lowe's Ratio 233
Number of Robust matches 59


 42%|████      | 21/50 [00:10<00:13,  2.17it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 294
Number of Robust matches 71


 44%|████      | 22/50 [00:10<00:12,  2.17it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 112
Number of Robust matches 29


 46%|████      | 23/50 [00:11<00:12,  2.16it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 200
Number of Robust matches 45


 48%|████      | 24/50 [00:11<00:12,  2.15it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 205
Number of Robust matches 38


 50%|█████     | 25/50 [00:12<00:11,  2.12it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 195
Number of Robust matches 36


 52%|█████     | 26/50 [00:12<00:11,  2.03it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 227
Number of Robust matches 32


 54%|█████     | 27/50 [00:13<00:15,  1.52it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 236
Number of Robust matches 33


 56%|█████     | 28/50 [00:14<00:14,  1.51it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 200
Number of Robust matches 30


 58%|█████     | 29/50 [00:14<00:12,  1.67it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 370
Number of Robust matches 48
```

60%|██████ | 30/50 [00:15<00:11, 1.77it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 283
Number of Robust matches 33
```

62%|██████ | 31/50 [00:15<00:10, 1.86it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 188
Number of Robust matches 49
```

64%|██████ | 32/50 [00:16<00:09, 1.97it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 183
Number of Robust matches 48
```

66%|██████ | 33/50 [00:16<00:08, 2.02it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 25
Number of Robust matches 6
```

68%|██████ | 34/50 [00:17<00:07, 2.04it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 63
Number of Robust matches 14
```

70%|██████ | 35/50 [00:17<00:07, 2.08it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 12
Number of Robust matches 4
```

72%|██████ | 36/50 [00:18<00:06, 2.07it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 105
Number of Robust matches 21
```

74%|██████ | 37/50 [00:18<00:06, 2.09it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 321
Number of Robust matches 48
```

76%|██████ | 38/50 [00:19<00:05, 2.08it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 252
Number of Robust matches 29
```

78%|██████ | 39/50 [00:19<00:05, 2.10it/s]

```
Number of matches 2000
Number of matches After Lowe's Ratio 178
Number of Robust matches 24


 80%|████████  | 40/50 [00:19<00:04,  2.08it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 201
Number of Robust matches 39


 82%|████████  | 41/50 [00:20<00:04,  2.05it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 131
Number of Robust matches 18


 84%|████████  | 42/50 [00:20<00:03,  2.05it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 334
Number of Robust matches 54


 86%|████████  | 43/50 [00:21<00:03,  2.05it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 132
Number of Robust matches 24


 88%|████████  | 44/50 [00:21<00:02,  2.10it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 148
Number of Robust matches 44


 90%|████████  | 45/50 [00:22<00:02,  2.07it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 262
Number of Robust matches 53


 92%|████████  | 46/50 [00:22<00:01,  2.09it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 135
Number of Robust matches 32


 94%|████████  | 47/50 [00:23<00:01,  2.16it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 66
Number of Robust matches 21


 96%|████████  | 48/50 [00:23<00:00,  2.15it/s]

Number of matches 2000
Number of matches After Lowe's Ratio 295
Number of Robust matches 67
```

```
 98%|█████████▊| 49/50 [00:24<00:00,  2.01it/s]
```

```
Number of matches 2000
Number of matches After Lowe's Ratio 182
Number of Robust matches 43
```

In [ ]:

```
H_left_gftt = []
H_right_gftt = []

num_matches_gftt = []
num_good_matches_gftt = []

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

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

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

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

In [ ]:

```
H_left_daisy = []
H_right_daisy = []

num_matches_daisy = []
num_good_matches_daisy = []

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

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

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

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

In [19]:

```
H_left_fast = []
H_right_fast = []

num_matches_fast = []
num_good_matches_fast = []

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

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

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

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

```
  2%|▏          | 1/51 [00:13<11:36, 13.94s/it]
```

```
Number of matches 87536
Number of matches After Lowe's Ratio 10285
Number of Robust matches 7414
```

```
  4%|▍          | 2/51 [00:31<12:55, 15.83s/it]
```

```
Number of matches 75857
Number of matches After Lowe's Ratio 644
Number of Robust matches 360
```

```
  6%|▌          | 3/51 [00:47<12:45, 15.95s/it]
```

```
Number of matches 97150
Number of matches After Lowe's Ratio 3315
Number of Robust matches 2144


Number of matches 98403
Number of matches After Lowe's Ratio 7816
```

```
  8%|▊          | 4/51 [01:06<13:31, 17.26s/it]
```

```
Number of Robust matches 5167
```

```
 10%|▉          | 5/51 [01:25<13:43, 17.91s/it]
```

```
Number of matches 94577
Number of matches After Lowe's Ratio 10068
Number of Robust matches 6903


Number of matches 94935
Number of matches After Lowe's Ratio 7153
```

```
 12%|          | 6/51 [01:44<13:41, 18.26s/it]

Number of Robust matches 5243


 14%|          | 7/51 [02:03<13:39, 18.63s/it]

Number of matches 101108
Number of matches After Lowe's Ratio 12097
Number of Robust matches 8939


Number of matches 103611
Number of matches After Lowe's Ratio 25654
 16%|          | 8/51 [02:25<13:55, 19.44s/it]

Number of Robust matches 21024


 18%|          | 9/51 [02:46<14:06, 20.15s/it]

Number of matches 106411
Number of matches After Lowe's Ratio 22967
Number of Robust matches 17258


 20%|          | 10/51 [03:08<14:04, 20.60s/it]

Number of matches 115231
Number of matches After Lowe's Ratio 14042
Number of Robust matches 9206


 22%|          | 11/51 [03:32<14:22, 21.57s/it]

Number of matches 117947
Number of matches After Lowe's Ratio 4381
Number of Robust matches 2563


Number of matches 111829
Number of matches After Lowe's Ratio 24831
 24%|          | 12/51 [03:55<14:27, 22.23s/it]

Number of Robust matches 17367


 25%|          | 13/51 [04:18<14:13, 22.47s/it]

Number of matches 112111
Number of matches After Lowe's Ratio 4896
Number of Robust matches 2316


 27%|          | 14/51 [04:41<13:53, 22.53s/it]

Number of matches 116212
Number of matches After Lowe's Ratio 18905
Number of Robust matches 11974


 29%|          | 15/51 [05:05<13:50, 23.08s/it]

Number of matches 123205
Number of matches After Lowe's Ratio 101
```

```
Number of Robust matches 16

 31%|███       | 16/51 [05:30<13:41, 23.47s/it]

Number of matches 109372
Number of matches After Lowe's Ratio 392
Number of Robust matches 119

 33%|███       | 17/51 [05:52<13:00, 22.96s/it]

Number of matches 112914
Number of matches After Lowe's Ratio 12118
Number of Robust matches 5780

 35%|███       | 18/51 [06:14<12:33, 22.83s/it]

Number of matches 109913
Number of matches After Lowe's Ratio 465
Number of Robust matches 118

 37%|███       | 19/51 [06:37<12:13, 22.93s/it]

Number of matches 120423
Number of matches After Lowe's Ratio 251
Number of Robust matches 73

 39%|███       | 20/51 [07:01<11:59, 23.22s/it]

Number of matches 121925
Number of matches After Lowe's Ratio 6830
Number of Robust matches 2945

 41%|████      | 21/51 [07:25<11:40, 23.35s/it]

Number of matches 116705
Number of matches After Lowe's Ratio 4466
Number of Robust matches 2003

 43%|████      | 22/51 [07:48<11:18, 23.39s/it]

Number of matches 109716
Number of matches After Lowe's Ratio 47
Number of Robust matches 8

 45%|████      | 23/51 [08:10<10:42, 22.95s/it]

Number of matches 110779
Number of matches After Lowe's Ratio 20404
Number of Robust matches 12165

 47%|████      | 24/51 [08:32<10:07, 22.52s/it]

Number of matches 109349
Number of matches After Lowe's Ratio 19120
Number of Robust matches 11205
```

```
 49%|████▉     | 25/51 [08:55<09:48, 22.62s/it]

Number of matches 115938
Number of matches After Lowe's Ratio 17206
Number of Robust matches 9056


 51%|█████     | 26/51 [09:18<09:30, 22.82s/it]

Number of matches 122855
Number of matches After Lowe's Ratio 14504
Number of Robust matches 7178


 53%|█████▎    | 27/51 [09:42<09:19, 23.31s/it]

Number of matches 133153
Number of matches After Lowe's Ratio 18941
Number of Robust matches 9616



Number of matches 139272
Number of matches After Lowe's Ratio 21770
Number of Robust matches 8807


 57%|█████▋    | 29/51 [10:36<09:13, 25.18s/it]

Number of matches 132310
Number of matches After Lowe's Ratio 20357
Number of Robust matches 11194


 59%|█████▉    | 30/51 [11:01<08:48, 25.17s/it]

Number of matches 117037
Number of matches After Lowe's Ratio 21976
Number of Robust matches 11928



Number of matches 112444
Number of matches After Lowe's Ratio 25813
 61%|██████    | 31/51 [11:25<08:14, 24.71s/it]

Number of Robust matches 14385



Number of matches 105817
Number of matches After Lowe's Ratio 25373
 63%|██████▎   | 32/51 [11:47<07:33, 23.85s/it]

Number of Robust matches 18998


 65%|██████▍   | 33/51 [12:08<06:55, 23.08s/it]

Number of matches 103385
Number of matches After Lowe's Ratio 22741
Number of Robust matches 16336


 67%|██████▋   | 34/51 [12:29<06:22, 22.49s/it]

Number of matches 109002
```

```
Number of matches After Lowe's Ratio 21629
Number of Robust matches 14253



Number of matches 112071
Number of matches After Lowe's Ratio 23284
Number of Robust matches 16528


 69%|██████▉   | 35/51 [12:52<06:00, 22.50s/it]

Number of matches 106802
Number of matches After Lowe's Ratio 25177
 71%|███████   | 36/51 [13:15<05:38, 22.58s/it]

Number of Robust matches 15680


 73%|███████▎  | 37/51 [13:36<05:13, 22.39s/it]

Number of matches 101920
Number of matches After Lowe's Ratio 18163
Number of Robust matches 11825


 75%|███████▌  | 38/51 [13:56<04:39, 21.53s/it]

Number of matches 85971
Number of matches After Lowe's Ratio 10146
Number of Robust matches 7521


 76%|███████▋  | 39/51 [14:14<04:04, 20.38s/it]

Number of matches 81836
Number of matches After Lowe's Ratio 23569
Number of Robust matches 18118


 78%|███████▊  | 40/51 [14:30<03:31, 19.23s/it]

Number of matches 87852
Number of matches After Lowe's Ratio 18363
Number of Robust matches 14441


 80%|████████  | 41/51 [14:48<03:08, 18.83s/it]

Number of matches 90007
Number of matches After Lowe's Ratio 18022
Number of Robust matches 11814


 82%|████████▏ | 42/51 [15:05<02:45, 18.36s/it]

Number of matches 89119
Number of matches After Lowe's Ratio 17660
Number of Robust matches 13178



Number of matches 93962
Number of matches After Lowe's Ratio 25922
Number of Robust matches 19786
```

```
 86%|████████▌ | 44/51 [15:42<02:08, 18.29s/it]
Number of matches 90558
Number of matches After Lowe's Ratio 7651
Number of Robust matches 5005


 88%|████████▊ | 45/51 [16:00<01:49, 18.28s/it]
Number of matches 89788
Number of matches After Lowe's Ratio 5179
Number of Robust matches 2932


Number of matches 94690
Number of matches After Lowe's Ratio 11661
 90%|████████▉ | 46/51 [16:19<01:31, 18.33s/it]
Number of Robust matches 5986


Number of matches 98372
Number of matches After Lowe's Ratio 10484
 92%|█████████▏| 47/51 [16:38<01:14, 18.65s/it]
Number of Robust matches 5047


Number of matches 97482
Number of matches After Lowe's Ratio 5874
 94%|█████████▍| 48/51 [16:58<00:57, 19.01s/it]
Number of Robust matches 2490


 96%|█████████▌| 49/51 [17:18<00:38, 19.24s/it]
Number of matches 100849
Number of matches After Lowe's Ratio 12345
Number of Robust matches 5167


 98%|█████████▊| 50/51 [17:37<00:21, 21.16s/it]
  0%|          | 0/50 [00:00<?, ?it/s]
Number of matches 92828
Number of matches After Lowe's Ratio 1114
Number of Robust matches 339


Number of matches 104129
Number of matches After Lowe's Ratio 6041
  2%|▏         | 1/50 [00:14<12:09, 14.89s/it]
Number of Robust matches 3455


  4%|▍         | 2/50 [00:34<14:17, 17.87s/it]
Number of matches 85258
Number of matches After Lowe's Ratio 5598
Number of Robust matches 3396
```

```
  6%|▌         | 3/50 [00:53<14:26, 18.44s/it]
Number of matches 113643
Number of matches After Lowe's Ratio 11815
Number of Robust matches 7010

  8%|▊         | 4/50 [01:16<15:21, 20.03s/it]
Number of matches 105684
Number of matches After Lowe's Ratio 2658
Number of Robust matches 1457

 10%|▉         | 5/50 [01:38<15:31, 20.69s/it]
Number of matches 109972
Number of matches After Lowe's Ratio 5694
Number of Robust matches 2745

 12%|█         | 6/50 [02:00<15:30, 21.14s/it]
Number of matches 107417
Number of matches After Lowe's Ratio 1195
Number of Robust matches 451

 14%|█▏        | 7/50 [02:21<15:11, 21.20s/it]
Number of matches 106987
Number of matches After Lowe's Ratio 15391
Number of Robust matches 7978

 16%|█▍        | 8/50 [02:43<15:03, 21.52s/it]
Number of matches 121549
Number of matches After Lowe's Ratio 106
Number of Robust matches 15

 18%|█▌        | 9/50 [03:07<15:05, 22.09s/it]
Number of matches 109090
Number of matches After Lowe's Ratio 1673
Number of Robust matches 669

 20%|█▊        | 10/50 [03:28<14:32, 21.81s/it]
Number of matches 105349
Number of matches After Lowe's Ratio 4230
Number of Robust matches 1866

 22%|█▉        | 11/50 [03:49<14:02, 21.59s/it]
Number of matches 108435
Number of matches After Lowe's Ratio 6737
Number of Robust matches 4144

 24%|██        | 12/50 [04:12<13:53, 21.93s/it]
```

```
Number of matches 123694
Number of matches After Lowe's Ratio 19791
Number of Robust matches 12013


 26%|██        | 13/50 [04:35<13:43, 22.25s/it]

Number of matches 96343
Number of matches After Lowe's Ratio 16656
Number of Robust matches 12393


 28%|██        | 14/50 [04:52<12:32, 20.90s/it]

Number of matches 54457
Number of matches After Lowe's Ratio 5962
Number of Robust matches 3988


 30%|███       | 15/50 [05:04<10:33, 18.11s/it]

Number of matches 74343
Number of matches After Lowe's Ratio 3528
Number of Robust matches 2306


 32%|███       | 16/50 [05:18<09:36, 16.95s/it]

Number of matches 57064
Number of matches After Lowe's Ratio 10178
Number of Robust matches 8030


 34%|███       | 17/50 [05:32<08:47, 15.97s/it]

Number of matches 104262
Number of matches After Lowe's Ratio 8765
Number of Robust matches 4524


 36%|███       | 18/50 [05:53<09:22, 17.57s/it]

Number of matches 105631
Number of matches After Lowe's Ratio 19903
Number of Robust matches 13545


 38%|███       | 19/50 [06:14<09:36, 18.61s/it]

Number of matches 108249
Number of matches After Lowe's Ratio 19738
Number of Robust matches 14067


 40%|████      | 20/50 [06:36<09:47, 19.59s/it]

Number of matches 106606
Number of matches After Lowe's Ratio 23842
Number of Robust matches 17248



Number of matches 120200
Number of matches After Lowe's Ratio 33402

 42%|████      | 21/50 [06:59<09:59, 20.66s/it]

Number of Robust matches 26874
```

Number of Robust matches 20071

44%|████ | 22/50 [07:24<10:12, 21.88s/it]

Number of matches 125528
Number of matches After Lowe's Ratio 9561
Number of Robust matches 5903

46%|████ | 23/50 [07:49<10:15, 22.79s/it]

Number of matches 129552
Number of matches After Lowe's Ratio 16308
Number of Robust matches 10996

48%|████ | 24/50 [08:15<10:19, 23.81s/it]

Number of matches 131203
Number of matches After Lowe's Ratio 18799
Number of Robust matches 12482

50%|█████ | 25/50 [08:41<10:11, 24.47s/it]

Number of matches 128349
Number of matches After Lowe's Ratio 14196
Number of Robust matches 8418

Number of matches 125112
Number of matches After Lowe's Ratio 9449

52%|█████ | 26/50 [09:07<09:56, 24.84s/it]

Number of Robust matches 4233

54%|█████ | 27/50 [09:31<09:25, 24.57s/it]

Number of matches 115397
Number of matches After Lowe's Ratio 18333
Number of Robust matches 8579

56%|█████ | 28/50 [09:54<08:49, 24.07s/it]

Number of matches 107757
Number of matches After Lowe's Ratio 14595
Number of Robust matches 7553

58%|█████ | 29/50 [10:16<08:12, 23.45s/it]

Number of matches 108520
Number of matches After Lowe's Ratio 21524
Number of Robust matches 10171

60%|██████ | 30/50 [10:37<07:36, 22.84s/it]

Number of matches 105170
Number of matches After Lowe's Ratio 21763
Number of Robust matches 10174

62%|██████ | 31/50 [10:58<07:03, 22.27s/it]

```
Number of matches 101842
Number of matches After Lowe's Ratio 6660
Number of Robust matches 3337
```

64%|██████ | 32/50 [11:19<06:33, 21.88s/it]

```
Number of matches 114806
Number of matches After Lowe's Ratio 7725
Number of Robust matches 3925
```

66%|██████ | 33/50 [11:43<06:22, 22.52s/it]

```
Number of matches 144158
Number of matches After Lowe's Ratio 2994
Number of Robust matches 1244
```

68%|██████▊ | 34/50 [12:11<06:24, 24.06s/it]

```
Number of matches 129327
Number of matches After Lowe's Ratio 13126
Number of Robust matches 4588
```

70%|███████ | 35/50 [12:38<06:12, 24.87s/it]

```
Number of matches 150305
Number of matches After Lowe's Ratio 52
Number of Robust matches 11
```

72%|███████ | 36/50 [13:06<06:02, 25.91s/it]

```
Number of matches 125780
Number of matches After Lowe's Ratio 10461
Number of Robust matches 3901
```

74%|███████▍ | 37/50 [13:31<05:33, 25.64s/it]

```
Number of matches 122865
Number of matches After Lowe's Ratio 7103
Number of Robust matches 2399
```

76%|███████▌ | 38/50 [13:54<04:59, 24.94s/it]

```
Number of matches 105783
Number of matches After Lowe's Ratio 17566
Number of Robust matches 8140
```

78%|███████▊ | 39/50 [14:15<04:21, 23.81s/it]

```
Number of matches 102138
Number of matches After Lowe's Ratio 5707
Number of Robust matches 2790
```

```
Number of matches 89671
Number of matches After Lowe's Ratio 10545
```

```
 80%|████████  | 40/50 [14:36<03:47, 22.72s/it]

Number of Robust matches 4038



Number of matches 96125
Number of matches After Lowe's Ratio 16071
 82%|████████  | 41/50 [14:54<03:11, 21.30s/it]

Number of Robust matches 6874



Number of matches 100177
Number of matches After Lowe's Ratio 26715
 84%|████████  | 42/50 [15:14<02:47, 20.94s/it]

Number of Robust matches 11299


 86%|████████  | 43/50 [15:34<02:25, 20.84s/it]

Number of matches 107031
Number of matches After Lowe's Ratio 13205
Number of Robust matches 6459


 88%|████████  | 44/50 [15:56<02:05, 21.00s/it]

Number of matches 103108
Number of matches After Lowe's Ratio 505
Number of Robust matches 153


 90%|█████████ | 45/50 [16:16<01:44, 20.95s/it]

Number of matches 107375
Number of matches After Lowe's Ratio 10404
Number of Robust matches 5760


 92%|█████████ | 46/50 [16:38<01:24, 21.04s/it]

Number of matches 108739
Number of matches After Lowe's Ratio 7698
Number of Robust matches 3892


 94%|█████████ | 47/50 [17:00<01:04, 21.46s/it]

Number of matches 116973
Number of matches After Lowe's Ratio 5070
Number of Robust matches 2995


 96%|█████████ | 48/50 [17:24<00:44, 22.10s/it]

Number of matches 116549
Number of matches After Lowe's Ratio 10498
Number of Robust matches 7584



Number of matches 107771
Number of matches After Lowe's Ratio 8033
 98%|█████████ | 49/50 [17:46<00:21, 21.77s/it]
```

```
Number of Robust matches 4890
```

In [ ]:

```
H_left_star = []
H_right_star = []

num_matches_star = []
num_good_matches_star = []

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

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

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

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

In [ ]:

```
H_left_sift = []
H_right_sift = []

num_matches_sift = []
num_good_matches_sift = []

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

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

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

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

In [ ]:

```
H_left_surf = []
H_right_surf = []
```

```
num_matches_surf = []
num_good_matches_surf = []

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

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

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

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

In [ ]:

```
H_left_surfsift = []
H_right_surfsift = []

num_matches_surfsift = []
num_good_matches_surfsift = []

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

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

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

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

In [20]:

```
H_left_agast = []
H_right_agast = []

num_matches_agast = []
num_good_matches_agast = []

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

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_agast[j:j+2][::-1],points_all_left_agast[j:j+2][::-1],descriptors_all_left_agast[j:j+2]
[::-1])
    H_left_agast.append(H_a)
```

```
    num_matches_agast.append(matches)
    num_good_matches_agast.append(gd_matches)

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

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_agast[j:j+2][::-1],points_all_right_agast[j:j+2][::-1],descriptors_all_right_agast[j:
j+2][::-1])
    H_right_agast.append(H_a)
    num_matches_agast.append(matches)
    num_good_matches_agast.append(gd_matches)
```

```
  0%|          | 0/51 [00:00<?, ?it/s]
```

```
Number of matches 98954
Number of matches After Lowe's Ratio 9380
```

```
  2%|          | 1/51 [00:15<13:06, 15.74s/it]
```

```
Number of Robust matches 6059
```

```
  4%|          | 2/51 [00:34<14:16, 17.48s/it]
```

```
Number of matches 79366
Number of matches After Lowe's Ratio 593
Number of Robust matches 304
```

```
  6%|          | 3/51 [00:51<13:56, 17.43s/it]
```

```
Number of matches 110204
Number of matches After Lowe's Ratio 3245
Number of Robust matches 2087
```

```
  8%|          | 4/51 [01:13<15:01, 19.19s/it]
```

```
Number of matches 105901
Number of matches After Lowe's Ratio 7215
Number of Robust matches 4947
```

```
 10%|          | 5/51 [01:34<15:09, 19.78s/it]
```

```
Number of matches 107504
Number of matches After Lowe's Ratio 9704
Number of Robust matches 7356
```

```
 12%|          | 6/51 [01:55<15:01, 20.02s/it]
```

```
Number of matches 103271
Number of matches After Lowe's Ratio 6633
Number of Robust matches 5082
```

```
 14%|          | 7/51 [02:16<15:07, 20.61s/it]
```

```
Number of matches 116933
Number of matches After Lowe's Ratio 11628
Number of Robust matches 9142
```

```
Number of matches 119028
Number of matches After Lowe's Ratio 25392
```

```
 16%|▉         | 8/51 [02:40<15:34, 21.74s/it]
Number of Robust matches 21361


Number of matches 124258
Number of matches After Lowe's Ratio 22155
 18%|▉         | 9/51 [03:05<15:43, 22.47s/it]
Number of Robust matches 13957

 20%|▉         | 10/51 [03:30<15:57, 23.35s/it]
Number of matches 131124
Number of matches After Lowe's Ratio 13375
Number of Robust matches 8887

 22%|▉         | 11/51 [03:56<16:09, 24.24s/it]
Number of matches 134054
Number of matches After Lowe's Ratio 4552
Number of Robust matches 2757


Number of matches 127609
Number of matches After Lowe's Ratio 24165
 24%|▉         | 12/51 [04:23<16:21, 25.17s/it]
Number of Robust matches 17371

 25%|▉         | 13/51 [04:49<16:01, 25.30s/it]
Number of matches 124545
Number of matches After Lowe's Ratio 4827
Number of Robust matches 2325


Number of matches 130795
Number of matches After Lowe's Ratio 17823
 27%|▉         | 14/51 [05:14<15:32, 25.19s/it]
Number of Robust matches 12487

 29%|▉         | 15/51 [05:41<15:24, 25.67s/it]
Number of matches 141804
Number of matches After Lowe's Ratio 98
Number of Robust matches 17

 31%|▉         | 16/51 [06:09<15:25, 26.43s/it]
Number of matches 128214
Number of matches After Lowe's Ratio 483
Number of Robust matches 138

 33%|▉         | 17/51 [06:35<14:53, 26.28s/it]
```

```
Number of matches 130355
Number of matches After Lowe's Ratio 11437
Number of Robust matches 5760
```

```
 35%|███       | 18/51 [07:00<14:19, 26.04s/it]
```

```
Number of matches 120660
Number of matches After Lowe's Ratio 463
Number of Robust matches 128
```

```
 37%|███▋      | 19/51 [07:25<13:41, 25.67s/it]
```

```
Number of matches 137824
Number of matches After Lowe's Ratio 263
Number of Robust matches 79
```

```
 39%|███▉      | 20/51 [07:52<13:30, 26.15s/it]
```

```
Number of matches 141614
Number of matches After Lowe's Ratio 6793
Number of Robust matches 2746
```

```
 41%|████      | 21/51 [08:20<13:15, 26.51s/it]
```

```
Number of matches 131535
Number of matches After Lowe's Ratio 4291
Number of Robust matches 1657
```

```
 43%|████▎     | 22/51 [08:46<12:45, 26.41s/it]
```

```
Number of matches 127536
Number of matches After Lowe's Ratio 58
Number of Robust matches 11
```

```
 45%|████▌     | 23/51 [09:11<12:11, 26.13s/it]
```

```
Number of matches 129839
Number of matches After Lowe's Ratio 20514
Number of Robust matches 12835
```

```
 47%|████▋     | 24/51 [09:37<11:42, 26.02s/it]
```

```
Number of matches 126651
Number of matches After Lowe's Ratio 18600
Number of Robust matches 11649
```

```
 49%|████▉     | 25/51 [10:02<11:09, 25.75s/it]
```

```
Number of matches 135662
Number of matches After Lowe's Ratio 16962
Number of Robust matches 10211
```

```
 51%|█████     | 26/51 [10:29<10:53, 26.15s/it]
```

```
Number of matches 139772
Number of matches After Lowe's Ratio 13798
Number of Robust matches 6939
```

```
 53%|█████     | 27/51 [10:59<10:53, 27.21s/it]
Number of matches 152790
Number of matches After Lowe's Ratio 18978
Number of Robust matches 9749


Number of matches 158880
Number of matches After Lowe's Ratio 21063
Number of Robust matches 9734

 55%|█████     | 28/51 [11:30<10:53, 28.39s/it]
Number of matches 152717
Number of matches After Lowe's Ratio 20181
 57%|█████     | 29/51 [12:01<10:41, 29.17s/it]
Number of Robust matches 9721

 59%|█████     | 30/51 [12:31<10:14, 29.28s/it]
Number of matches 132257
Number of matches After Lowe's Ratio 21394
Number of Robust matches 11899


Number of matches 126012
Number of matches After Lowe's Ratio 25535
 61%|██████    | 31/51 [12:58<09:33, 28.67s/it]
Number of Robust matches 15492


Number of matches 118729
Number of matches After Lowe's Ratio 25260
Number of Robust matches 17911

 65%|██████    | 33/51 [13:46<07:50, 26.16s/it]
Number of matches 112501
Number of matches After Lowe's Ratio 21357
Number of Robust matches 12944

 67%|██████    | 34/51 [14:09<07:09, 25.29s/it]
Number of matches 120367
Number of matches After Lowe's Ratio 21552
Number of Robust matches 14063


Number of matches 128900
Number of matches After Lowe's Ratio 22874
 69%|██████    | 35/51 [14:35<06:49, 25.62s/it]
Number of Robust matches 15481
```

```
Number of matches 118239
Number of matches After Lowe's Ratio 24094
Number of Robust matches 14131


 73%|███████   | 37/51 [15:25<05:52, 25.15s/it]

Number of matches 117746
Number of matches After Lowe's Ratio 17921
Number of Robust matches 11227


 75%|███████   | 38/51 [15:48<05:18, 24.50s/it]

Number of matches 94610
Number of matches After Lowe's Ratio 9216
Number of Robust matches 6658


Number of matches 93123
Number of matches After Lowe's Ratio 22665
 76%|███████   | 39/51 [16:07<04:35, 22.98s/it]

Number of Robust matches 14114


 78%|███████   | 40/51 [16:26<03:58, 21.66s/it]

Number of matches 94525
Number of matches After Lowe's Ratio 16661
Number of Robust matches 9543


 80%|████████  | 41/51 [16:45<03:28, 20.83s/it]

Number of matches 97507
Number of matches After Lowe's Ratio 17477
Number of Robust matches 11025


Number of matches 93618
Number of matches After Lowe's Ratio 16473
 82%|████████  | 42/51 [17:04<03:02, 20.30s/it]

Number of Robust matches 11280


Number of matches 100100
Number of matches After Lowe's Ratio 24989
 84%|████████  | 43/51 [17:23<02:40, 20.01s/it]

Number of Robust matches 16093


 86%|████████  | 44/51 [17:43<02:19, 20.00s/it]

Number of matches 102617
Number of matches After Lowe's Ratio 7552
Number of Robust matches 4608


Number of matches 103996
Number of matches After Lowe's Ratio 5151
```

```
 88%|████████▊ | 45/51 [18:03<02:00, 20.04s/it]
Number of Robust matches 2985

 90%|█████████ | 46/51 [18:25<01:42, 20.48s/it]
Number of matches 112006
Number of matches After Lowe's Ratio 11259
Number of Robust matches 5926

 92%|█████████▏| 47/51 [18:47<01:23, 21.00s/it]
Number of matches 111932
Number of matches After Lowe's Ratio 9440
Number of Robust matches 4507

 94%|█████████▍| 48/51 [19:10<01:04, 21.64s/it]
Number of matches 115210
Number of matches After Lowe's Ratio 5950
Number of Robust matches 2380

 96%|█████████▌| 49/51 [19:34<00:44, 22.19s/it]
Number of matches 116566
Number of matches After Lowe's Ratio 11728
Number of Robust matches 6179

 98%|█████████▊| 50/51 [19:57<00:23, 23.95s/it]
  0%|          | 0/50 [00:00<?, ?it/s]
Number of matches 108059
Number of matches After Lowe's Ratio 1188
Number of Robust matches 423

  2%|▏         | 1/50 [00:16<13:13, 16.19s/it]
Number of matches 113929
Number of matches After Lowe's Ratio 5928
Number of Robust matches 3305

  4%|▍         | 2/50 [00:38<15:54, 19.89s/it]
Number of matches 100181
Number of matches After Lowe's Ratio 5535
Number of Robust matches 3058

  6%|▌         | 3/50 [01:00<16:10, 20.65s/it]
Number of matches 127159
Number of matches After Lowe's Ratio 11330
Number of Robust matches 6738

  8%|▊         | 4/50 [01:26<17:28, 22.79s/it]
Number of matches 120719
Number of matches After Lowe's Ratio 2706
```

```
Number of Robust matches 1335


 10%|▉         | 5/50 [01:51<17:40, 23.57s/it]

Number of matches 127483
Number of matches After Lowe's Ratio 5828
Number of Robust matches 2950


 12%|▉         | 6/50 [02:16<17:45, 24.21s/it]

Number of matches 119944
Number of matches After Lowe's Ratio 1346
Number of Robust matches 550


 14%|▉         | 7/50 [02:41<17:26, 24.34s/it]

Number of matches 119233
Number of matches After Lowe's Ratio 15219
Number of Robust matches 9118


 16%|▉         | 8/50 [03:06<17:08, 24.49s/it]

Number of matches 135893
Number of matches After Lowe's Ratio 109
Number of Robust matches 16


 18%|▉         | 9/50 [03:32<17:09, 25.10s/it]

Number of matches 123626
Number of matches After Lowe's Ratio 1806
Number of Robust matches 687


 20%|▉         | 10/50 [03:57<16:37, 24.94s/it]

Number of matches 117876
Number of matches After Lowe's Ratio 3970
Number of Robust matches 1919


 22%|▉         | 11/50 [04:21<16:02, 24.68s/it]

Number of matches 120007
Number of matches After Lowe's Ratio 6810
Number of Robust matches 3838


Number of matches 136160
Number of matches After Lowe's Ratio 19054

 24%|▉         | 12/50 [04:45<15:37, 24.67s/it]

Number of Robust matches 12081


Number of matches 106595
Number of matches After Lowe's Ratio 15909

 26%|▉         | 13/50 [05:12<15:29, 25.12s/it]

Number of Robust matches 12150
```

```
 28%|██        | 14/50 [05:31<13:57, 23.28s/it]

Number of matches 58345
Number of matches After Lowe's Ratio 5456
Number of Robust matches 3889


 30%|███       | 15/50 [05:43<11:42, 20.08s/it]

Number of matches 81499
Number of matches After Lowe's Ratio 3471
Number of Robust matches 1966


 32%|███       | 16/50 [05:59<10:36, 18.73s/it]

Number of matches 63079
Number of matches After Lowe's Ratio 9610
Number of Robust matches 6701


 34%|███       | 17/50 [06:14<09:45, 17.74s/it]

Number of matches 114291
Number of matches After Lowe's Ratio 8634
Number of Robust matches 4714


 36%|███       | 18/50 [06:38<10:23, 19.48s/it]

Number of matches 116170
Number of matches After Lowe's Ratio 18935
Number of Robust matches 11406


 38%|███       | 19/50 [07:01<10:35, 20.50s/it]

Number of matches 119668
Number of matches After Lowe's Ratio 18911
Number of Robust matches 12735


 40%|████      | 20/50 [07:25<10:52, 21.75s/it]

Number of matches 117998
Number of matches After Lowe's Ratio 22979
Number of Robust matches 16462


Number of matches 132866
Number of matches After Lowe's Ratio 33225
 42%|████      | 21/50 [07:51<11:06, 22.98s/it]

Number of Robust matches 22450


 44%|████      | 22/50 [08:19<11:22, 24.36s/it]

Number of matches 140126
Number of matches After Lowe's Ratio 9334
Number of Robust matches 6913


 46%|████      | 23/50 [08:47<11:27, 25.48s/it]
```

Number of matches 141469
Number of matches After Lowe's Ratio 15281
Number of Robust matches 10019

48%|████▊ | 24/50 [09:15<11:24, 26.32s/it]

Number of matches 145304
Number of matches After Lowe's Ratio 18170
Number of Robust matches 9201

50%|█████ | 25/50 [09:43<11:12, 26.88s/it]

Number of matches 140550
Number of matches After Lowe's Ratio 12908
Number of Robust matches 6371

52%|█████▏ | 26/50 [10:11<10:53, 27.23s/it]

Number of matches 145499
Number of matches After Lowe's Ratio 9533
Number of Robust matches 3913

54%|█████▍ | 27/50 [10:41<10:39, 27.81s/it]

Number of matches 131418
Number of matches After Lowe's Ratio 16844
Number of Robust matches 7725

56%|█████▌ | 28/50 [11:06<09:57, 27.14s/it]

Number of matches 118887
Number of matches After Lowe's Ratio 13539
Number of Robust matches 6862

58%|█████▊ | 29/50 [11:30<09:10, 26.20s/it]

Number of matches 120713
Number of matches After Lowe's Ratio 20902
Number of Robust matches 9338

60%|██████ | 30/50 [11:54<08:29, 25.48s/it]

Number of matches 119094
Number of matches After Lowe's Ratio 21504
Number of Robust matches 9107

Number of matches 110691
Number of matches After Lowe's Ratio 6409

62%|██████▏ | 31/50 [12:17<07:52, 24.86s/it]

Number of Robust matches 3214

64%|██████▍ | 32/50 [12:40<07:16, 24.24s/it]

Number of matches 128357

Number of matches After Lowe's Ratio 8043
Number of Robust matches 4140

66%|██████▌ | 33/50 [13:07<07:04, 24.96s/it]

Number of matches 157198
Number of matches After Lowe's Ratio 3021
Number of Robust matches 1039

68%|██████▊ | 34/50 [13:37<07:06, 26.66s/it]

Number of matches 141400
Number of matches After Lowe's Ratio 13136
Number of Robust matches 5338

70%|███████ | 35/50 [14:05<06:46, 27.09s/it]

Number of matches 162834
Number of matches After Lowe's Ratio 36
Number of Robust matches 8

72%|███████▏ | 36/50 [14:37<06:36, 28.31s/it]

Number of matches 136771
Number of matches After Lowe's Ratio 10597
Number of Robust matches 4613

74%|███████▍ | 37/50 [15:04<06:02, 27.90s/it]

Number of matches 139861
Number of matches After Lowe's Ratio 7004
Number of Robust matches 2436

76%|███████▌ | 38/50 [15:31<05:32, 27.72s/it]

Number of matches 121764
Number of matches After Lowe's Ratio 16901
Number of Robust matches 7137

78%|███████▊ | 39/50 [15:55<04:51, 26.54s/it]

Number of matches 110405
Number of matches After Lowe's Ratio 5286
Number of Robust matches 2552

80%|████████ | 40/50 [16:15<04:07, 24.75s/it]

Number of matches 96942
Number of matches After Lowe's Ratio 10306
Number of Robust matches 3413

82%|████████▏ | 41/50 [16:35<03:30, 23.34s/it]

Number of matches 106417
Number of matches After Lowe's Ratio 15769
Number of Robust matches 6164

```
Number of matches 110856
Number of matches After Lowe's Ratio 26934
 84%|████████▍ | 42/50 [16:57<03:03, 22.95s/it]
Number of Robust matches 11975


 86%|████████▌ | 43/50 [17:20<02:39, 22.78s/it]
Number of matches 119085
Number of matches After Lowe's Ratio 13437
Number of Robust matches 6113


 88%|████████▊ | 44/50 [17:43<02:18, 23.03s/it]
Number of matches 119392
Number of matches After Lowe's Ratio 580
Number of Robust matches 179


 90%|█████████ | 45/50 [18:08<01:57, 23.45s/it]
Number of matches 128047
Number of matches After Lowe's Ratio 10055
Number of Robust matches 5162


 92%|█████████▏ | 46/50 [18:33<01:35, 23.90s/it]
Number of matches 121827
Number of matches After Lowe's Ratio 7500
Number of Robust matches 4169


 94%|█████████▍ | 47/50 [18:57<01:11, 23.99s/it]
Number of matches 133957
Number of matches After Lowe's Ratio 5160
Number of Robust matches 3112


 96%|█████████▌ | 48/50 [19:24<00:49, 24.98s/it]
Number of matches 132546
Number of matches After Lowe's Ratio 9959
Number of Robust matches 6884


 98%|█████████▊ | 49/50 [19:50<00:24, 24.30s/it]
Number of matches 122243
Number of matches After Lowe's Ratio 7812
Number of Robust matches 4973
```

In [20]:

```
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 [21]:

```
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 [27]:

```
xmax,xmin,ymax,ymin,t,h,w,Ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_superpoint,H_right_superpoint)
```

```
Step1:Done
Step2:Done
```

In [ ]:

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

In [ ]:

```
warp_imgs_all_star = final_step_right_union(warp_imgs_left,images_right_bgr_no_enhance,H_
right_star,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

In [ ]:

```
xmax,xmin,ymax,ymin,t,h,w,Ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_star,H_right_star)
```

In [ ]:

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

In [ ]:

```
warp_imgs_all_star = final_step_right_union(warp_imgs_left,images_right_bgr_no_enhance,H_
right_star,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

In [ ]:

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

In [ ]:

```
omax,omin,umax,umin,T,H,W,HT = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_sift,H_right_sift)
```

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

```
warp_imgs_all_fast = final_step_right_union(warp_imgs_fast,images_right_bgr_no_enhance,H_
right_fast,mmax,mmin,nmax,nmin,d,e,f,g)
```

In [ ]:

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

In [24]:

```
omax,omin,umax,umin,T,H,W,HT = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_akaze,H_right_akaze)
```

```
Step1:Done
Step2:Done
```

In [ ]:

```
warp_img_kaze = final_steps_left_union(images_left_bgr_no_enhance,H_left_akaze,omax,omin,
umax,umin,T,H,W,HT)
```

In [ ]:

```
warp_imgs_all_akaze = final_step_right_union(warp_img_kaze,images_right_bgr_no_enhance,H_
right_akaze,omax,omin,umax,umin,T,H,W,HT)
```

In [ ]:

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

In [22]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_freak,H_right_freak)
```

```
Step1:Done
Step2:Done
```

In [23]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_freak,amax,ami
n,zmax,zmin,d,i,q,ht)
```

```
step31:Done
```

In [24]:

```
warp_imgs_all_gftt = final_step_right_union(warp_image_left,images_right_bgr_no_enhance,H
_right_freak,amax,amin,zmax,zmin,d,i,q,ht)
```

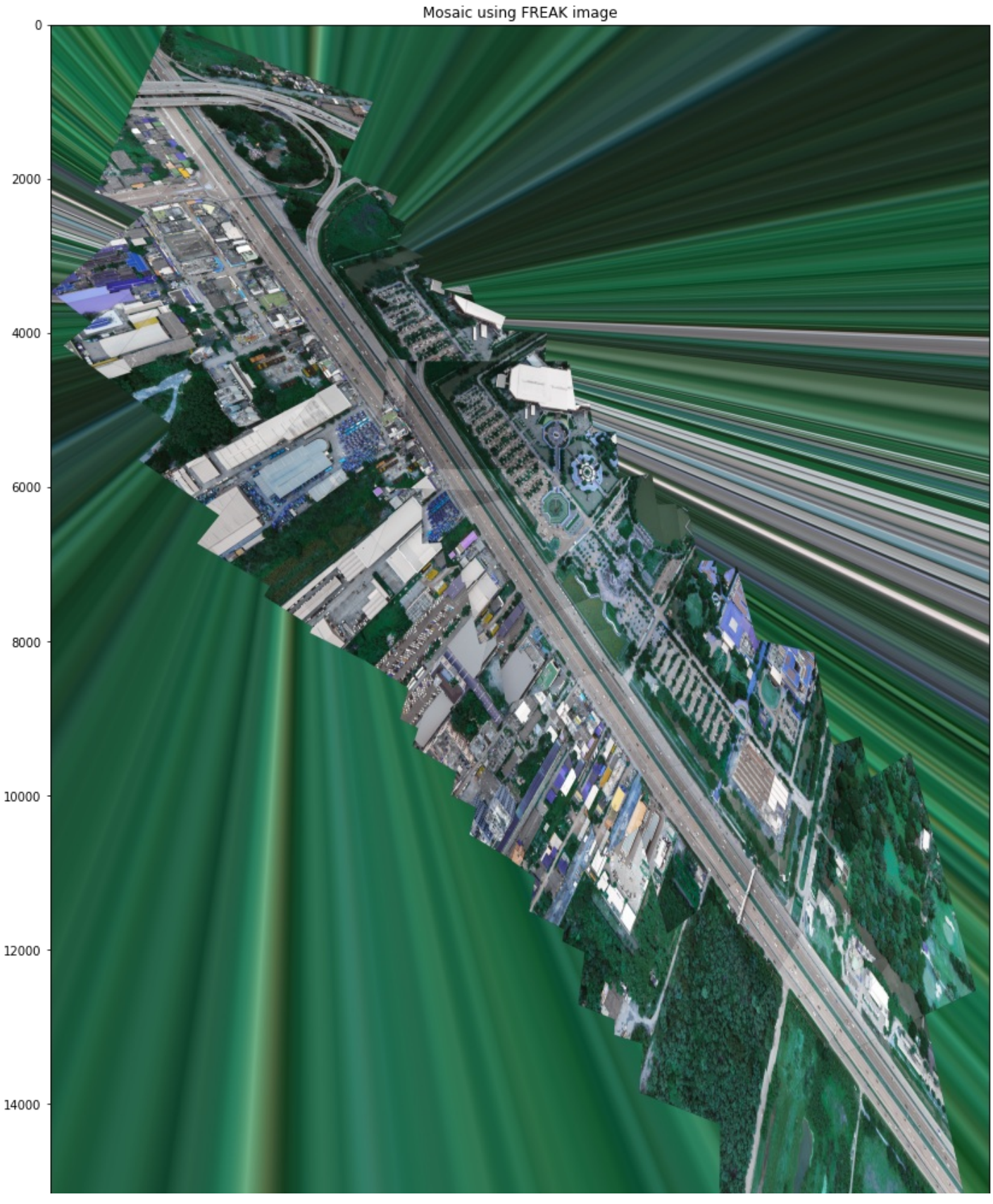
```
step32:Done
```

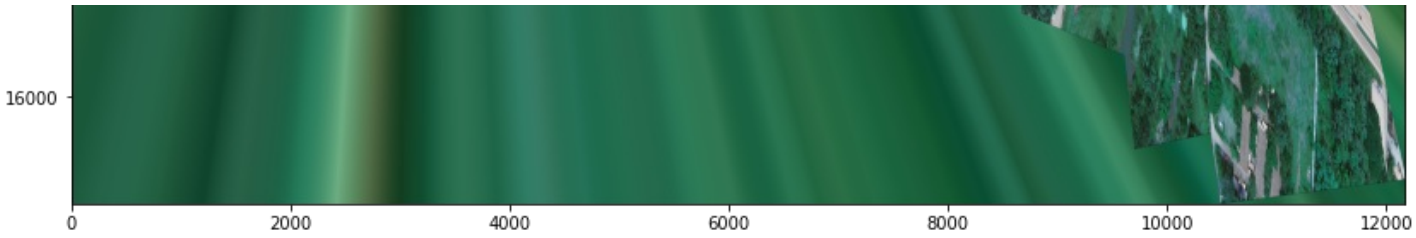
In [26]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_gftt)
plt.title('Mosaic using FREAK image')
```

Out[26]:

```
Text(0.5, 1.0, 'Mosaic using FREAK image')
```





In [22]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_fast,H_right_fast)
```

```
Step1:Done
Step2:Done
```

In [ ]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_fast,amax,amin
,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
warp_imgs_all_agast = final_step_right_union(warp_image_left,images_right_bgr_no_enhance,
H_right_fast,amax,amin,zmax,zmin,d,i,q,ht)
```

In [ ]:

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

In [23]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_agast,H_right_agast)
```

```
Step1:Done
Step2:Done
```

In [ ]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_agast,amax,ami
n,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
warp_imgs_all_agast = final_step_right_union(warp_image_left,images_right_bgr_no_enhance,
H_right_agast,amax,amin,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
plt.figure(figsize=(20,20))
plt.imshow(warp_imgs_all_agast)
plt.title('Mosaic using AGAST image')
```

In [ ]:

```

```