

```
In [12]:
import numpy as np
import cv2
import scipy.io
import os
from numpy.linalg import norm
from matplotlib import pyplot as plt
from numpy.linalg import det
from numpy.linalg import inv
from scipy.linalg import rq
from numpy.linalg import svd
import matplotlib.pyplot as plt
import numpy as np
import math
import random
import sys
from scipy import ndimage, spatial
from tqdm.notebook import tqdm, trange

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 numpy as np
import math
import glob
import matplotlib.pyplot as plt
import time
import os
import copy
import sklearn.svm
import cv2
from matplotlib import pyplot as plt
import numpy as np
from os.path import exists
import pandas as pd
import PIL
import random
from google.colab import drive
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 [13]:
from google.colab import drive
# This will prompt for authorization.
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

```
In [14]:
!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 /usr/local/lib/python3.7/dist-packages (3.4.2.17)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-python==3.4.2.17) (1.19.5)
Requirement already satisfied: opencv-contrib-python==3.4.2.17 in /usr/local/lib/python3.7/dist-packages (3.4.2.17)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-contrib-python==3.4.2.17) (1.19.5)
```

```
In [15]:
class Image:
    def __init__(self, img, position):
        self.img = img
        self.position = position

    inlier_matchset = []
    def features_matching(a, keypointlength, threshold):
        #threshold=0.2
        bestmatch=np.empty((keypointlength),dtype= np.int16)
        img1Index=np.empty((keypointlength),dtype=np.int16)
        distance=np.empty((keypointlength))
        index=0
        for j in range(0,keypointlength):
            #For a descriptor fa in Ia, take the two closest descriptors fb1 and fb2 in Ib
            x=[j]
            listx=x.tolist()
            x.sort()
            minval=x[0] # min
            minval=x[1] # 2nd min
            itemindex1 = listx.index(minval) #index of min val
            itemindex2 = listx.index(minval2) #Index of second min value
            ratio=minval1/minval2 #Ratio test

            if ratio

```

```
def compute_Homography(im1_pts,im2_pts):
    """
    im1_pts and im2_pts are 2xn matrices with
    4 point correspondences from the two images
    """
    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] # First row
        row2 = [0, 0, 0, a_x, a_y, 1, -b_y*a_x, -b_y*a_y, -b_y] # Second row
        # place the rows in the matrix
        A[A_index] = row1
        A[A_index+1] = row2
```

```

a_index += 2

U, s, Vt = np.linalg.svd(A)

#s is a 1-D array of singular values sorted in descending order
#U, Vt are unitary matrices
#Rows of Vt are the eigenvectors of A^T A.
#Columns of U are the eigenvectors of A A^T.
H = np.eye(3)
H = Vt[-1].reshape(3,3) # take the last row of the Vt matrix
return H

```

```

def displayplot(img,title):

    plt.figure(figsize(15,15))
    plt.title(title)
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    plt.show()

```

```

In [16]: def get_inliers(f1, f2, matches, H, RANSACthresh):

    inlier_indices = []
    for i in range(len(matches)):
        queryInd = matches[i].queryIdx
        trainInd = matches[i].trainIdx

        #queryInd = matches[i][0]
        #trainInd = matches[i][1]

        queryPoint = np.array([f1[queryInd].pt[0], f1[queryInd].pt[1], 1]).T
        trans_query = H.dot(queryPoint)

        comp1 = [trans_query[0]/trans_query[2], trans_query[1]/trans_query[2]] # normalize with respect to z
        comp2 = np.array(f2[trainInd].pt)[2]

        if(np.linalg.norm(comp1-comp2) <= RANSACthresh): # check against threshold
            inlier_indices.append(i)
    return inlier_indices

def RANSAC_alg(f1, f2, matches, nRANSAC, RANSACthresh):

    minMatches = 4
    nBest = 0
    best_inliers = []
    H_estimate = np.eye(3,3)
    global inlier_matchset
    inlier_matchset=[]
    for iteration in range(nRANSAC):

        #Choose a minimal set of feature matches.
        matchSample = random.sample(matches, minMatches)

        #Estimate the Homography implied by these matches
        im1_pts=np.empty((minMatches,2))
        im2_pts=np.empty((minMatches,2))
        for i in range(0,minMatches):
            m = matchSample[i]
            im1_pts[i] = f1[m.queryIdx].pt
            im2_pts[i] = f2[m.trainIdx].pt
            #im1_pts[i] = f1[m[0]].pt
            #im2_pts[i] = f2[m[1]].pt

        H_estimate=compute_Homography(im1_pts,im2_pts)

        # Calculate the inliers for the H
        inliers = get_inliers(f1, f2, matches, H_estimate, RANSACthresh)

        # if the number of inliers is higher than previous iterations, update the best estimates
        if len(inliers) > nBest:
            nBest= len(inliers)
            best_inliers = inliers

    print("Number of best inliers",len(best_inliers))
    for i in range(len(best_inliers)):
        inlier_matchset.append(matches[best_inliers[i]])

    # compute a homography given this set of matches
    im1_pts=np.empty(((len(best_inliers),2)))
    im2_pts=np.empty(((len(best_inliers),2)))
    for i in range(0,len(best_inliers)):
        m = inlier_matchset[i]
        im1_pts[i] = f1[m.queryIdx].pt
        im2_pts[i] = f2[m.trainIdx].pt
        #im1_pts[i] = f1[m[0]].pt
        #im2_pts[i] = f2[m[1]].pt

    M=compute_Homography(im1_pts,im2_pts)
    return M, best_inliers

```

```

In [17]: files_all=[]
for file in os.listdir("/content/drive/MyDrive/Aerial/"):
    if file.endswith(".JPG"):
        files_all.append(file)

files_all.sort()
folder_path = '/content/drive/MyDrive/Aerial/'

centre_file = folder_path + files_all[60]
left_files_path_rev = []
right_files_path = []

for file in files_all[:61]:
    left_files_path_rev.append(folder_path + file)

left_files_path = left_files_path_rev[::-1]

for file in files_all[59:120]:
    right_files_path.append(folder_path + file)

```

```

In [18]: from PIL.ExifTags import TAGS
from PIL.ExifTags import GPSTAGS
from PIL import Image
def get_exif(filename):

```

```

image = Image.open(filename)
image.verify()
return image._getexif()

def get_labeled_exif(exif):
    labeled = {}
    for (key, val) in exif.items():
        labeled[TAGS.get(key)] = val

    return labeled

def get_geotagging(exif):
    if not exif:
        raise ValueError("No EXIF metadata found")

    geotagging = {}
    for (idx, tag) in TAGS.items():
        if tag == "GPSInfo":
            if idx not in exif:
                raise ValueError("No EXIF geotagging found")

            for (key, val) in GPSTAGS.items():
                if key in exif[idx]:
                    geotagging[val] = exif[idx][key]
    return geotagging

def get_decimal_from_dms(dms, ref):
    degrees = dms[0][0] / dms[0][1]
    minutes = dms[1][0] / dms[1][1] / 60.0
    seconds = dms[2][0] / dms[2][1] / 3600.0

    if ref in ['S', 'W']:
        degrees = -degrees
        minutes = -minutes
        seconds = -seconds

    return round(degrees + minutes + seconds, 5)

def get_coordinates(geotags):
    lat = get_decimal_from_dms(geotags['GPSLatitude'], geotags['GPSLatitudeRef'])
    lon = get_decimal_from_dms(geotags['GPSLongitude'], geotags['GPSLongitudeRef'])

    return (lat, lon)

```

In [19]: gridsize = 8

```

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(gridsize,gridsize))

images_left_bgr = []
images_right_bgr = []

images_left = []
images_right = []

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

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

```

100% |██████████| 61/61 [01:10<00:00, 1.15s/it]

In [20]:

```

images_left_bgr_no_enhance = []
images_right_bgr_no_enhance = []

for file in tqdm(left_files_path):
    left_image_sat= cv2.imread(file)
    left_img = cv2.resize(left_image_sat, None, fx=0.35, fy=0.35, interpolation = cv2.INTER_CUBIC)
    images_left_bgr_no_enhance.append(left_img)

for file in tqdm(right_files_path):
    right_image_sat= cv2.imread(file)
    right_img = cv2.resize(right_image_sat, None, fx=0.35, fy=0.35, interpolation = cv2.INTER_CUBIC)
    images_right_bgr_no_enhance.append(right_img)

```

100% |██████████| 61/61 [00:26<00:00, 2.28it/s]

In [21]:

```

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, desc) = 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
        desc /= (np.linalg.norm(descs, axis=0, ord=2) + eps)
        desc /= (descs.sum(axis=0) + eps)
        desc = np.sqrt(descs)
        desc /= (np.linalg.norm(descs, axis=0, ord=2) + eps)

        # return a tuple of the keypoints and descriptors
        return (kps, desc)

```

In [22]:

```
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% |██████████| 61/61 [02:52<00:00, 2.83s/it]  
100% |██████████| 61/61 [02:48<00:00, 2.76s/it]

```

In [23]: 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):
    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):
    kpt = sift.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_sift.append(kpt)
    descriptors_all_right_sift.append(descrip)
    points_all_right_sift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

100% |██████████| 61/61 [02:50<00:00, 2.80s/it]  
100% |██████████| 61/61 [02:49<00:00, 2.78s/it]

```

In [24]: 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, ransacReprojThreshold =thresh)

    inliers = inliers.flatten()
    return H, inliers
```

```

In [25]: def compute_homography_fast_other(matched_pts1, matched_pts2):
    #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,
                                    0)
    inliers = inliers.flatten()
    return H, inliers
```

```

In [26]: 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_1.append((m[0].trainIdx, m[0].queryIdx))
            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])
    matches_idx = np.array([m.trainIdx for m in matches_4])
    imm2_pts = np.array([keypts[1][idx].pt for idx in matches_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, RANSACThresh=6)
```

```

...
Hn,inliers = compute_homography_fast(imm1_pts,imm2_pts,thresh)
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_1f1_1f:
        # 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)

#globally 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_1f1_1f), len(inlier_matchset)

```

```

In [27]: from functools import partial
from tqdm import tqdm
tqdm = partial(tqdm, position=0, leave=True)

In [28]: H_left_rootsift = []
H_right_rootsift = []

num_matches_rootsift = []
num_good_matches_rootsift = []

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_rootsift[j:j+2][::-1],points_all_left_rootsift[j:j+2][::-1],descriptors_all_left_rootsift[j:j+2][::-1])
    H_left_rootsift.append(H_a)
    num_matches_rootsift.append(matches)
    num_good_matches_rootsift.append(gd_matches)

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

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_rootsift[j:j+2][::-1],points_all_right_rootsift[j:j+2][::-1],descriptors_all_right_rootsift[j:j+2][::-1])
    H_right_rootsift.append(H_a)
    #num_matches.append(matches)
    #num_good_matches.append(gd_matches)

2%|██████████| 1/61 [00:06<06:05,  6.10s/it]
Number of matches 28871
Number of matches After Lowe's Ratio 2376
Number of Robust matches 1265

3%|████| 2/61 [00:12<06:01,  6.13s/it]
Number of matches 35330
Number of matches After Lowe's Ratio 1612
Number of Robust matches 757

5%|████| 3/61 [00:19<06:17,  6.52s/it]
Number of matches 32332
Number of matches After Lowe's Ratio 608
Number of Robust matches 250

7%|████| 4/61 [00:26<06:13,  6.56s/it]
Number of matches 32125
Number of matches After Lowe's Ratio 4741
Number of Robust matches 2564

8%|████| 5/61 [00:33<06:11,  6.64s/it]
Number of matches 32463
Number of matches After Lowe's Ratio 5263
Number of Robust matches 3131

10%|████| 6/61 [00:39<06:07,  6.68s/it]
Number of matches 32266
Number of matches After Lowe's Ratio 4938
Number of Robust matches 2861

Number of matches 32877
Number of matches After Lowe's Ratio 5332
11%|████| 7/61 [00:46<06:01,  6.69s/it]
Number of Robust matches 3103

13%|████| 8/61 [00:53<05:51,  6.64s/it]
Number of matches 27613
Number of matches After Lowe's Ratio 3048
Number of Robust matches 1978

15%|████| 9/61 [00:58<05:28,  6.31s/it]

```

Number of matches 31966  
Number of matches After Lowe's Ratio 4287  
Number of Robust matches 2754

16% | [ 10/61 [01:04<05:18, 6.24s/it]  
Number of matches 23666  
Number of matches After Lowe's Ratio 1868  
Number of Robust matches 1251

18% | [ 11/61 [01:09<04:49, 5.79s/it]  
Number of matches 27909  
Number of matches After Lowe's Ratio 4022  
Number of Robust matches 2652

20% | [ 12/61 [01:14<04:35, 5.62s/it]  
Number of matches 24162  
Number of matches After Lowe's Ratio 3911  
Number of Robust matches 2583

21% | [ 13/61 [01:19<04:15, 5.33s/it]  
Number of matches 27592  
Number of matches After Lowe's Ratio 4142  
Number of Robust matches 3014

23% | [ 14/61 [01:24<04:13, 5.39s/it]  
Number of matches 27282  
Number of matches After Lowe's Ratio 5930  
Number of Robust matches 4576

25% | [ 15/61 [01:30<04:06, 5.35s/it]  
Number of matches 27099  
Number of matches After Lowe's Ratio 4364  
Number of Robust matches 3214

26% | [ 16/61 [01:35<03:57, 5.27s/it]  
Number of matches 25919  
Number of matches After Lowe's Ratio 4589  
Number of Robust matches 3399

28% | [ 17/61 [01:40<03:49, 5.21s/it]  
Number of matches 26581  
Number of matches After Lowe's Ratio 4326  
Number of Robust matches 3436

30% | [ 18/61 [01:45<03:42, 5.18s/it]  
Number of matches 27052  
Number of matches After Lowe's Ratio 5515  
Number of Robust matches 4258

31% | [ 19/61 [01:50<03:39, 5.22s/it]  
Number of matches 28182  
Number of matches After Lowe's Ratio 6206  
Number of Robust matches 4974

33% | [ 20/61 [01:56<03:38, 5.33s/it]  
Number of matches 29913  
Number of matches After Lowe's Ratio 5138  
Number of Robust matches 4158

34% | [ 21/61 [02:02<03:45, 5.65s/it]  
Number of matches 32208  
Number of matches After Lowe's Ratio 4678  
Number of Robust matches 2908

36% | [ 22/61 [02:09<03:56, 6.07s/it]  
Number of matches 31209  
Number of matches After Lowe's Ratio 4729  
Number of Robust matches 2928

38% | [ 23/61 [02:16<03:56, 6.23s/it]  
Number of matches 31666  
Number of matches After Lowe's Ratio 5049  
Number of Robust matches 3262

39% | [ 24/61 [02:23<03:58, 6.44s/it]  
Number of matches 31039  
Number of matches After Lowe's Ratio 4088  
Number of Robust matches 2500

41% | [ 25/61 [02:30<03:56, 6.57s/it]  
Number of matches 35893  
Number of matches After Lowe's Ratio 3420  
Number of Robust matches 1687

43% | [ 26/61 [02:38<04:03, 6.95s/it]  
Number of matches 30633  
Number of matches After Lowe's Ratio 3630  
Number of Robust matches 2040

44% | [ 27/61 [02:45<03:57, 6.98s/it]  
Number of matches 35656  
Number of matches After Lowe's Ratio 3585  
Number of Robust matches 2051

46% | [ 28/61 [02:52<03:57, 7.21s/it]  
Number of matches 34271  
Number of matches After Lowe's Ratio 3970  
Number of Robust matches 1682

48% | [29/61 [03:01<03:59, 7.50s/it]

Number of matches 39115

Number of matches After Lowe's Ratio 2483

Number of Robust matches 1091

49% | [30/61 [03:10<04:08, 8.01s/it]

Number of matches 38049

Number of matches After Lowe's Ratio 4210

Number of Robust matches 2148

51% | [31/61 [03:18<04:05, 8.19s/it]

Number of matches 38187

Number of matches After Lowe's Ratio 2051

Number of Robust matches 936

52% | [32/61 [03:27<04:01, 8.32s/it]

Number of matches 35025

Number of matches After Lowe's Ratio 873

Number of Robust matches 386

54% | [33/61 [03:34<03:45, 8.05s/it]

Number of matches 35065

Number of matches After Lowe's Ratio 3787

Number of Robust matches 2279

56% | [34/61 [03:42<03:33, 7.90s/it]

Number of matches 32733

Number of matches After Lowe's Ratio 4252

Number of Robust matches 2558

57% | [35/61 [03:49<03:17, 7.59s/it]

Number of matches 33968

Number of matches After Lowe's Ratio 3815

Number of Robust matches 2543

59% | [36/61 [03:56<03:10, 7.61s/it]

Number of matches 39070

Number of matches After Lowe's Ratio 4956

Number of Robust matches 2427

61% | [37/61 [04:06<03:13, 8.08s/it]

Number of matches 41604

Number of matches After Lowe's Ratio 4663

Number of Robust matches 1956

62% | [38/61 [04:16<03:19, 8.68s/it]

Number of matches 43881

Number of matches After Lowe's Ratio 4707

Number of Robust matches 1961

64% | [39/61 [04:26<03:23, 9.24s/it]

Number of matches 40123

Number of matches After Lowe's Ratio 4328

Number of Robust matches 2186

66% | [40/61 [04:36<03:16, 9.37s/it]

Number of matches 36596

Number of matches After Lowe's Ratio 4843

Number of Robust matches 2525

67% | [41/61 [04:45<03:02, 9.14s/it]

Number of matches 33986

Number of matches After Lowe's Ratio 5091

Number of Robust matches 3036

69% | [42/61 [04:52<02:41, 8.50s/it]

Number of matches 30652

Number of matches After Lowe's Ratio 5010

Number of Robust matches 3025

70% | [43/61 [04:58<02:20, 7.80s/it]

Number of matches 29868

Number of matches After Lowe's Ratio 5565

Number of Robust matches 3506

72% | [44/61 [05:04<02:02, 7.20s/it]

Number of matches 28588

Number of matches After Lowe's Ratio 4586

Number of Robust matches 3429

Number of matches 28661

Number of matches After Lowe's Ratio 5213

74% | [45/61 [05:09<01:48, 6.75s/it]

Number of Robust matches 3097

75% | [46/61 [05:15<01:35, 6.37s/it]

Number of matches 27785

Number of matches After Lowe's Ratio 5653

Number of Robust matches 3994

77% | [47/61 [05:20<01:24, 6.06s/it]

Number of matches 27261

Number of matches After Lowe's Ratio 5481

Number of Robust matches 3649

79% | [48/61 [05:25<01:15, 5.79s/it]

Number of matches 23943

Number of matches After Lowe's Ratio 3329

Number of Robust matches 2213

80% | [ ] 49/61 [05:30<01:04, 5.39s/it]  
Number of matches 23206  
Number of matches After Lowe's Ratio 5662  
Number of Robust matches 4300

82% | [ ] 50/61 [05:34<00:56, 5.10s/it]  
Number of matches 27133  
Number of matches After Lowe's Ratio 5256  
Number of Robust matches 4203

84% | [ ] 51/61 [05:40<00:52, 5.21s/it]  
Number of matches 28897  
Number of matches After Lowe's Ratio 4089  
Number of Robust matches 2977

85% | [ ] 52/61 [05:46<00:51, 5.70s/it]  
Number of matches 30356  
Number of matches After Lowe's Ratio 4285  
Number of Robust matches 2997

87% | [ ] 53/61 [05:53<00:48, 6.12s/it]  
Number of matches 38770  
Number of matches After Lowe's Ratio 5543  
Number of Robust matches 4002

89% | [ ] 54/61 [06:01<00:45, 6.47s/it]  
Number of matches 30376  
Number of matches After Lowe's Ratio 4359  
Number of Robust matches 2789

90% | [ ] 55/61 [06:08<00:39, 6.64s/it]  
Number of matches 30122  
Number of matches After Lowe's Ratio 4666  
Number of Robust matches 2986

92% | [ ] 56/61 [06:15<00:33, 6.75s/it]  
Number of matches 30579  
Number of matches After Lowe's Ratio 4053  
Number of Robust matches 2322

93% | [ ] 57/61 [06:22<00:27, 6.89s/it]  
Number of matches 29556  
Number of matches After Lowe's Ratio 5821  
Number of Robust matches 3148

95% | [ ] 58/61 [06:29<00:20, 6.91s/it]  
Number of matches 30860  
Number of matches After Lowe's Ratio 3318  
Number of Robust matches 1547

97% | [ ] 59/61 [06:36<00:14, 7.06s/it]  
Number of matches 30396  
Number of matches After Lowe's Ratio 5245  
Number of Robust matches 2090

0% | [ ] 0/61 [00:00<?, ?it/s]  
Number of matches 26655  
Number of matches After Lowe's Ratio 1603  
Number of Robust matches 615

2% | [ ] 1/61 [00:05<05:52, 5.87s/it]  
Number of matches 30330  
Number of matches After Lowe's Ratio 2350  
Number of Robust matches 1440

3% | [ ] 2/61 [00:13<06:17, 6.39s/it]  
Number of matches 32312  
Number of matches After Lowe's Ratio 2420  
Number of Robust matches 1410

5% | [ ] 3/61 [00:21<06:35, 6.82s/it]  
Number of matches 32696  
Number of matches After Lowe's Ratio 3958  
Number of Robust matches 2660

7% | [ ] 4/61 [00:29<06:46, 7.13s/it]  
Number of matches 31067  
Number of matches After Lowe's Ratio 4163  
Number of Robust matches 3011

8% | [ ] 5/61 [00:35<06:27, 6.92s/it]  
Number of matches 19735  
Number of matches After Lowe's Ratio 928  
Number of Robust matches 729

10% | [ ] 6/61 [00:39<05:29, 6.00s/it]  
Number of matches 27026  
Number of matches After Lowe's Ratio 1119  
Number of Robust matches 637

11% | [ ] 7/61 [00:44<05:05, 5.66s/it]  
Number of matches 19200  
Number of matches After Lowe's Ratio 2591  
Number of Robust matches 2068

13% | [ ] 8/61 [00:48<04:29, 5.08s/it]  
Number of matches 28248  
Number of matches After Lowe's Ratio 1741

Number of Robust matches 1248

15% | 9/61 [00:54<04:44, 5.48s/it]

Number of matches 29428

Number of matches After Lowe's Ratio 5864

Number of Robust matches 4613

16% | 10/61 [01:00<04:44, 5.58s/it]

Number of matches 28737

Number of matches After Lowe's Ratio 5928

Number of Robust matches 4672

Number of matches 28036

Number of matches After Lowe's Ratio 4909

18% | 11/61 [01:06<04:42, 5.64s/it]

Number of Robust matches 3208

20% | 12/61 [01:11<04:36, 5.64s/it]

Number of matches 31014

Number of matches After Lowe's Ratio 5069

Number of Robust matches 4206

21% | 13/61 [01:19<05:02, 6.29s/it]

Number of matches 34946

Number of matches After Lowe's Ratio 3977

Number of Robust matches 2567

23% | 14/61 [01:28<05:33, 7.10s/it]

Number of matches 35939

Number of matches After Lowe's Ratio 5314

Number of Robust matches 3674

25% | 15/61 [01:37<05:53, 7.68s/it]

Number of matches 36837

Number of matches After Lowe's Ratio 4817

Number of Robust matches 2718

26% | 16/61 [01:46<06:09, 8.21s/it]

Number of matches 37856

Number of matches After Lowe's Ratio 5229

Number of Robust matches 2540

28% | 17/61 [01:56<06:18, 8.61s/it]

Number of matches 37941

Number of matches After Lowe's Ratio 5295

Number of Robust matches 2217

30% | 18/61 [02:05<06:17, 8.78s/it]

Number of matches 34580

Number of matches After Lowe's Ratio 5114

Number of Robust matches 2438

31% | 19/61 [02:14<06:05, 8.70s/it]

Number of matches 33372

Number of matches After Lowe's Ratio 3992

Number of Robust matches 1664

Number of matches 33599

Number of matches After Lowe's Ratio 5710

33% | 20/61 [02:21<05:40, 8.30s/it]

Number of Robust matches 2614

34% | 21/61 [02:28<05:19, 8.00s/it]

Number of matches 34878

Number of matches After Lowe's Ratio 4827

Number of Robust matches 2830

36% | 22/61 [02:36<05:06, 7.85s/it]

Number of matches 32992

Number of matches After Lowe's Ratio 4726

Number of Robust matches 2128

38% | 23/61 [02:43<04:50, 7.63s/it]

Number of matches 34056

Number of matches After Lowe's Ratio 3744

Number of Robust matches 1834

39% | 24/61 [02:51<04:47, 7.76s/it]

Number of matches 44723

Number of matches After Lowe's Ratio 1513

Number of Robust matches 617

41% | 25/61 [03:02<05:17, 8.82s/it]

Number of matches 41727

Number of matches After Lowe's Ratio 3055

Number of Robust matches 1448

43% | 26/61 [03:13<05:23, 9.24s/it]

Number of matches 46928

Number of matches After Lowe's Ratio 466

Number of Robust matches 155

44% | 27/61 [03:24<05:36, 9.90s/it]

Number of matches 39889

Number of matches After Lowe's Ratio 2687

Number of Robust matches 1084

46% | [28/61 [03:33<05:18, 9.65s/it]  
Number of matches 38874  
Number of matches After Lowe's Ratio 4915  
Number of Robust matches 1941

Number of matches 33707  
Number of matches After Lowe's Ratio 3709  
48% | [29/61 [03:42<05:00, 9.38s/it]  
Number of Robust matches 1395

49% | [30/61 [03:49<04:32, 8.78s/it]  
Number of matches 32337  
Number of matches After Lowe's Ratio 3250  
Number of Robust matches 1397

51% | [31/61 [03:56<04:05, 8.18s/it]  
Number of matches 31634  
Number of matches After Lowe's Ratio 3434  
Number of Robust matches 1256

52% | [32/61 [04:03<03:46, 7.83s/it]  
Number of matches 31779  
Number of matches After Lowe's Ratio 3910  
Number of Robust matches 1658

54% | [33/61 [04:10<03:31, 7.55s/it]  
Number of matches 32533  
Number of matches After Lowe's Ratio 6121  
Number of Robust matches 2739

56% | [34/61 [04:17<03:21, 7.46s/it]  
Number of matches 34689  
Number of matches After Lowe's Ratio 3728  
Number of Robust matches 1663

57% | [35/61 [04:25<03:16, 7.56s/it]  
Number of matches 32975  
Number of matches After Lowe's Ratio 5852  
Number of Robust matches 2889

59% | [36/61 [04:32<03:05, 7.41s/it]  
Number of matches 32080  
Number of matches After Lowe's Ratio 4582  
Number of Robust matches 2525

61% | [37/61 [04:39<02:56, 7.34s/it]  
Number of matches 32233  
Number of matches After Lowe's Ratio 3719  
Number of Robust matches 2329

62% | [38/61 [04:46<02:45, 7.18s/it]  
Number of matches 30573  
Number of matches After Lowe's Ratio 3295  
Number of Robust matches 2659

64% | [39/61 [04:52<02:32, 6.92s/it]  
Number of matches 30768  
Number of matches After Lowe's Ratio 3999  
Number of Robust matches 2472

66% | [40/61 [04:59<02:21, 6.73s/it]  
Number of matches 26813  
Number of matches After Lowe's Ratio 3030  
Number of Robust matches 2812

67% | [41/61 [05:04<02:05, 6.30s/it]  
Number of matches 26443  
Number of matches After Lowe's Ratio 4773  
Number of Robust matches 3297

69% | [42/61 [05:09<01:54, 6.02s/it]  
Number of matches 28580  
Number of matches After Lowe's Ratio 2839  
Number of Robust matches 1673

70% | [43/61 [05:16<01:50, 6.12s/it]  
Number of matches 28143  
Number of matches After Lowe's Ratio 5319  
Number of Robust matches 3737

72% | [44/61 [05:21<01:38, 5.81s/it]  
Number of matches 17004  
Number of matches After Lowe's Ratio 1323  
Number of Robust matches 989

74% | [45/61 [05:23<01:18, 4.92s/it]  
Number of matches 13636  
Number of matches After Lowe's Ratio 2480  
Number of Robust matches 1655

75% | [46/61 [05:26<01:00, 4.06s/it]  
Number of matches 13604  
Number of matches After Lowe's Ratio 1412  
Number of Robust matches 813

77% | [47/61 [05:28<00:48, 3.47s/it]  
Number of matches 16334  
Number of matches After Lowe's Ratio 3288  
Number of Robust matches 2386

79% | [ ] 48/61 [05:31<00:44, 3.39s/it]

Number of matches 28245  
Number of matches After Lowe's Ratio 1401  
Number of Robust matches 1023

80% | [ ] 49/61 [05:37<00:50, 4.21s/it]

Number of matches 31687  
Number of matches After Lowe's Ratio 4671  
Number of Robust matches 2912

Number of matches 33859  
Number of matches After Lowe's Ratio 4336  
82% | [ ] 50/61 [05:44<00:56, 5.10s/it]  
Number of Robust matches 2959

84% | [ ] 51/61 [05:51<00:56, 5.69s/it]

Number of matches 28095  
Number of matches After Lowe's Ratio 3557  
Number of Robust matches 2211

85% | [ ] 52/61 [05:57<00:51, 5.69s/it]

Number of matches 27389  
Number of matches After Lowe's Ratio 4188  
Number of Robust matches 2428

87% | [ ] 53/61 [06:03<00:46, 5.80s/it]

Number of matches 32426  
Number of matches After Lowe's Ratio 3116  
Number of Robust matches 1787

89% | [ ] 54/61 [06:10<00:42, 6.10s/it]

Number of matches 30438  
Number of matches After Lowe's Ratio 1294  
Number of Robust matches 600

90% | [ ] 55/61 [06:16<00:36, 6.13s/it]

Number of matches 28440  
Number of matches After Lowe's Ratio 2560  
Number of Robust matches 1640

92% | [ ] 56/61 [06:22<00:30, 6.05s/it]

Number of matches 26824  
Number of matches After Lowe's Ratio 4612  
Number of Robust matches 2605

93% | [ ] 57/61 [06:27<00:23, 5.86s/it]

Number of matches 26741  
Number of matches After Lowe's Ratio 2714  
Number of Robust matches 1559

95% | [ ] 58/61 [06:32<00:16, 5.64s/it]

Number of matches 22819  
Number of matches After Lowe's Ratio 3693  
Number of Robust matches 2666

97% | [ ] 59/61 [06:36<00:10, 5.12s/it]

Number of matches 17913  
Number of matches After Lowe's Ratio 3442  
Number of Robust matches 2493

98% | [ ] 60/61 [06:39<00:04, 4.52s/it]

Number of matches 16676  
Number of matches After Lowe's Ratio 3591  
Number of Robust matches 2731

```
In [29]: 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],0.5)
    H_left_sift.append(H_a)
    num_matches_sift.append(matches)
    num_good_matches_sift.append(gd_matches)

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

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_sift[j:j+2][::-1],points_all_right_sift[j:j+2][::-1],descriptors_all_right_sift[j:j+2][::-1],0.5)
    H_right_sift.append(H_a)
    #num_matches.append(matches)
    #num_good_matches.append(gd_matches)
```

2% | [ ] 1/61 [00:06<06:30, 6.51s/it]

Number of matches 28871  
Number of matches After Lowe's Ratio 570  
Number of Robust matches 458

3% | [ ] 2/61 [00:13<06:24, 6.52s/it]

Number of matches 35330  
Number of matches After Lowe's Ratio 377  
Number of Robust matches 252

5% | [ ] 3/61 [00:20<06:43, 6.95s/it]

Number of matches 32332  
Number of matches After Lowe's Ratio 67  
Number of Robust matches 59

7% | 4/61 [00:28<06:39, 7.00s/it]  
Number of matches 32125  
Number of matches After Lowe's Ratio 1490  
Number of Robust matches 985

8% | 5/61 [00:35<06:33, 7.03s/it]  
Number of matches 32463  
Number of matches After Lowe's Ratio 1704  
Number of Robust matches 875

10% | 6/61 [00:42<06:30, 7.10s/it]  
Number of matches 32266  
Number of matches After Lowe's Ratio 1665  
Number of Robust matches 1030

11% | 7/61 [00:49<06:20, 7.05s/it]  
Number of matches 32877  
Number of matches After Lowe's Ratio 1822  
Number of Robust matches 1145

13% | 8/61 [00:56<06:09, 6.97s/it]  
Number of matches 27613  
Number of matches After Lowe's Ratio 743  
Number of Robust matches 480

15% | 9/61 [01:02<05:52, 6.77s/it]  
Number of matches 31966  
Number of matches After Lowe's Ratio 1280  
Number of Robust matches 988

16% | 10/61 [01:08<05:38, 6.63s/it]  
Number of matches 23666  
Number of matches After Lowe's Ratio 506  
Number of Robust matches 417

18% | 11/61 [01:13<05:04, 6.09s/it]  
Number of matches 27909  
Number of matches After Lowe's Ratio 1297  
Number of Robust matches 1003

20% | 12/61 [01:19<04:48, 5.89s/it]  
Number of matches 24162  
Number of matches After Lowe's Ratio 1429  
Number of Robust matches 1103

21% | 13/61 [01:24<04:30, 5.63s/it]  
Number of matches 27592  
Number of matches After Lowe's Ratio 1426  
Number of Robust matches 1214

23% | 14/61 [01:29<04:24, 5.63s/it]  
Number of matches 27282  
Number of matches After Lowe's Ratio 2452  
Number of Robust matches 1893

25% | 15/61 [01:35<04:18, 5.62s/it]  
Number of matches 27099  
Number of matches After Lowe's Ratio 1669  
Number of Robust matches 1429

26% | 16/61 [01:40<04:10, 5.57s/it]  
Number of matches 25919  
Number of matches After Lowe's Ratio 2043  
Number of Robust matches 1888

28% | 17/61 [01:45<04:00, 5.47s/it]  
Number of matches 26581  
Number of matches After Lowe's Ratio 1911  
Number of Robust matches 1657

30% | 18/61 [01:51<03:53, 5.43s/it]  
Number of matches 27052  
Number of matches After Lowe's Ratio 2778  
Number of Robust matches 2614

31% | 19/61 [01:57<03:51, 5.51s/it]  
Number of matches 28182  
Number of matches After Lowe's Ratio 3182  
Number of Robust matches 2719

33% | 20/61 [02:02<03:51, 5.64s/it]  
Number of matches 29913  
Number of matches After Lowe's Ratio 2423  
Number of Robust matches 1717

34% | 21/61 [02:09<03:56, 5.91s/it]  
Number of matches 32208  
Number of matches After Lowe's Ratio 1922  
Number of Robust matches 1578

36% | 22/61 [02:16<04:03, 6.23s/it]  
Number of matches 31209  
Number of matches After Lowe's Ratio 1597  
Number of Robust matches 1279

38% | 23/61 [02:23<03:59, 6.31s/it]

Number of matches 31666  
Number of matches After Lowe's Ratio 1765  
Number of Robust matches 1465

39% | 24/61 [02:29<03:59, 6.46s/it]

Number of matches 31039  
Number of matches After Lowe's Ratio 1263  
Number of Robust matches 1042

41% | 25/61 [02:36<03:58, 6.61s/it]

Number of matches 35893  
Number of matches After Lowe's Ratio 1022  
Number of Robust matches 711

43% | 26/61 [02:44<04:04, 6.98s/it]

Number of matches 30633  
Number of matches After Lowe's Ratio 1066  
Number of Robust matches 792

44% | 27/61 [02:51<03:56, 6.95s/it]

Number of matches 35656  
Number of matches After Lowe's Ratio 1092  
Number of Robust matches 813

46% | 28/61 [02:59<03:59, 7.26s/it]

Number of matches 34271  
Number of matches After Lowe's Ratio 1077  
Number of Robust matches 657

48% | 29/61 [03:07<03:59, 7.49s/it]

Number of matches 39115  
Number of matches After Lowe's Ratio 598  
Number of Robust matches 370

49% | 30/61 [03:17<04:12, 8.16s/it]

Number of matches 38049  
Number of matches After Lowe's Ratio 1098  
Number of Robust matches 795

51% | 31/61 [03:26<04:12, 8.41s/it]

Number of matches 38187  
Number of matches After Lowe's Ratio 440  
Number of Robust matches 277

52% | 32/61 [03:34<04:05, 8.47s/it]

Number of matches 35825  
Number of matches After Lowe's Ratio 135  
Number of Robust matches 92

54% | 33/61 [03:42<03:54, 8.36s/it]

Number of matches 35065  
Number of matches After Lowe's Ratio 1075  
Number of Robust matches 771

56% | 34/61 [03:50<03:39, 8.14s/it]

Number of matches 32733  
Number of matches After Lowe's Ratio 1236  
Number of Robust matches 955

57% | 35/61 [03:57<03:24, 7.88s/it]

Number of matches 33968  
Number of matches After Lowe's Ratio 1165  
Number of Robust matches 844

59% | 36/61 [04:05<03:18, 7.95s/it]

Number of matches 39070  
Number of matches After Lowe's Ratio 1524  
Number of Robust matches 1215

61% | 37/61 [04:15<03:21, 8.39s/it]

Number of matches 41604  
Number of matches After Lowe's Ratio 1083  
Number of Robust matches 729

62% | 38/61 [04:25<03:26, 8.96s/it]

Number of matches 43881  
Number of matches After Lowe's Ratio 1007  
Number of Robust matches 581

64% | 39/61 [04:35<03:24, 9.32s/it]

Number of matches 40123  
Number of matches After Lowe's Ratio 940  
Number of Robust matches 636

66% | 40/61 [04:45<03:17, 9.41s/it]

Number of matches 36596  
Number of matches After Lowe's Ratio 1192  
Number of Robust matches 838

67% | 41/61 [04:53<03:00, 9.03s/it]

Number of matches 33986  
Number of matches After Lowe's Ratio 1616  
Number of Robust matches 1318

69% | 42/61 [05:00<02:39, 8.38s/it]

Number of matches 30652  
Number of matches After Lowe's Ratio 1769  
Number of Robust matches 1467

70% | [ ] 43/61 [05:06<02:19, 7.76s/it]  
Number of matches 29868  
Number of matches After Lowe's Ratio 2031  
Number of Robust matches 1662

72% | [ ] 44/61 [05:12<02:02, 7.19s/it]  
Number of matches 28588  
Number of matches After Lowe's Ratio 1944  
Number of Robust matches 1603

74% | [ ] 45/61 [05:18<01:48, 6.78s/it]  
Number of matches 28661  
Number of matches After Lowe's Ratio 2366  
Number of Robust matches 1780

75% | [ ] 46/61 [05:24<01:38, 6.55s/it]  
Number of matches 27785  
Number of matches After Lowe's Ratio 2587  
Number of Robust matches 1912

77% | [ ] 47/61 [05:30<01:27, 6.28s/it]  
Number of matches 27261  
Number of matches After Lowe's Ratio 2551  
Number of Robust matches 1781

79% | [ ] 48/61 [05:35<01:18, 6.01s/it]  
Number of matches 23943  
Number of matches After Lowe's Ratio 1441  
Number of Robust matches 1131

80% | [ ] 49/61 [05:40<01:07, 5.61s/it]  
Number of matches 23206  
Number of matches After Lowe's Ratio 2796  
Number of Robust matches 2264

82% | [ ] 50/61 [05:44<00:58, 5.29s/it]  
Number of matches 27133  
Number of matches After Lowe's Ratio 2504  
Number of Robust matches 2212

84% | [ ] 51/61 [05:50<00:54, 5.42s/it]  
Number of matches 28897  
Number of matches After Lowe's Ratio 1411  
Number of Robust matches 1222

85% | [ ] 52/61 [05:56<00:50, 5.59s/it]  
Number of matches 30356  
Number of matches After Lowe's Ratio 1362  
Number of Robust matches 1139

87% | [ ] 53/61 [06:02<00:46, 5.85s/it]  
Number of matches 30770  
Number of matches After Lowe's Ratio 1953  
Number of Robust matches 1512

89% | [ ] 54/61 [06:09<00:42, 6.05s/it]  
Number of matches 30376  
Number of matches After Lowe's Ratio 1317  
Number of Robust matches 838

90% | [ ] 55/61 [06:15<00:36, 6.10s/it]  
Number of matches 30122  
Number of matches After Lowe's Ratio 1662  
Number of Robust matches 1289

92% | [ ] 56/61 [06:22<00:31, 6.21s/it]  
Number of matches 30579  
Number of matches After Lowe's Ratio 1473  
Number of Robust matches 946

93% | [ ] 57/61 [06:28<00:24, 6.25s/it]  
Number of matches 29556  
Number of matches After Lowe's Ratio 2211  
Number of Robust matches 1183

95% | [ ] 58/61 [06:34<00:18, 6.19s/it]  
Number of matches 30860  
Number of matches After Lowe's Ratio 1150  
Number of Robust matches 564

97% | [ ] 59/61 [06:40<00:12, 6.29s/it]  
Number of matches 30396  
Number of matches After Lowe's Ratio 1897  
Number of Robust matches 977

98% | [ ] 60/61 [06:46<00:06, 6.20s/it]  
Number of matches 26655  
Number of matches After Lowe's Ratio 450  
Number of Robust matches 236

99% | [ ] 61/61 [06:52<00:00, 6.19s/it]  
Number of matches 30330  
Number of matches After Lowe's Ratio 560  
Number of Robust matches 467

100% | [ ] 62/61 [00:00<06:00, 6.00s/it]  
Number of matches 32312  
Number of matches After Lowe's Ratio 726

Number of Robust matches 600

5% | 3/61 [00:19<06:12, 6.42s/it]  
Number of matches 32696  
Number of matches After Lowe's Ratio 1303  
Number of Robust matches 1085

7% | 4/61 [00:26<06:15, 6.59s/it]  
Number of matches 31067  
Number of matches After Lowe's Ratio 1383  
Number of Robust matches 1208

8% | 5/61 [00:32<05:55, 6.35s/it]  
Number of matches 19735  
Number of matches After Lowe's Ratio 412  
Number of Robust matches 365

10% | 6/61 [00:36<05:10, 5.64s/it]  
Number of matches 27026  
Number of matches After Lowe's Ratio 333  
Number of Robust matches 272

11% | 7/61 [00:41<04:52, 5.41s/it]  
Number of matches 19200  
Number of matches After Lowe's Ratio 1276  
Number of Robust matches 1141

13% | 8/61 [00:45<04:20, 4.91s/it]  
Number of matches 28248  
Number of matches After Lowe's Ratio 662  
Number of Robust matches 532

15% | 9/61 [00:50<04:30, 5.20s/it]  
Number of matches 29428  
Number of matches After Lowe's Ratio 2610  
Number of Robust matches 2120

16% | 10/61 [00:56<04:36, 5.42s/it]  
Number of matches 28737  
Number of matches After Lowe's Ratio 2940  
Number of Robust matches 2657

18% | 11/61 [01:02<04:38, 5.57s/it]  
Number of matches 28036  
Number of matches After Lowe's Ratio 2496  
Number of Robust matches 2275

20% | 12/61 [01:08<04:39, 5.70s/it]  
Number of matches 31014  
Number of matches After Lowe's Ratio 2409  
Number of Robust matches 1928

21% | 13/61 [01:15<04:48, 6.01s/it]  
Number of matches 34946  
Number of matches After Lowe's Ratio 1782  
Number of Robust matches 1543

23% | 14/61 [01:23<05:08, 6.56s/it]  
Number of matches 35939  
Number of matches After Lowe's Ratio 2194  
Number of Robust matches 1912

25% | 15/61 [01:31<05:25, 7.09s/it]  
Number of matches 36837  
Number of matches After Lowe's Ratio 1631  
Number of Robust matches 1365

26% | 16/61 [01:39<05:34, 7.42s/it]  
Number of matches 37856  
Number of matches After Lowe's Ratio 1641  
Number of Robust matches 944

28% | 17/61 [01:48<05:41, 7.76s/it]  
Number of matches 37941  
Number of matches After Lowe's Ratio 1716  
Number of Robust matches 1159

30% | 18/61 [01:56<05:37, 7.84s/it]  
Number of matches 34580  
Number of matches After Lowe's Ratio 1869  
Number of Robust matches 1229

31% | 19/61 [02:03<05:24, 7.73s/it]  
Number of matches 33372  
Number of matches After Lowe's Ratio 1535  
Number of Robust matches 1023

33% | 20/61 [02:11<05:11, 7.59s/it]  
Number of matches 33599  
Number of matches After Lowe's Ratio 2155  
Number of Robust matches 1270

34% | 21/61 [02:18<05:00, 7.52s/it]  
Number of matches 34878  
Number of matches After Lowe's Ratio 1747  
Number of Robust matches 865

36% | 22/61 [02:26<04:55, 7.57s/it]

Number of matches 32992  
Number of matches After Lowe's Ratio 1691  
Number of Robust matches 932

38% | [23/61 [02:33<04:45, 7.50s/it]]  
Number of matches 34056  
Number of matches After Lowe's Ratio 1148  
Number of Robust matches 774

39% | [24/61 [02:41<04:45, 7.70s/it]]  
Number of matches 44723  
Number of matches After Lowe's Ratio 284  
Number of Robust matches 170

41% | [25/61 [02:52<05:14, 8.73s/it]]  
Number of matches 41727  
Number of matches After Lowe's Ratio 516  
Number of Robust matches 260

43% | [26/61 [03:03<05:21, 9.18s/it]]  
Number of matches 46928  
Number of matches After Lowe's Ratio 44  
Number of Robust matches 43

44% | [27/61 [03:14<05:32, 9.78s/it]]  
Number of matches 39889  
Number of matches After Lowe's Ratio 541  
Number of Robust matches 280

46% | [28/61 [03:23<05:14, 9.52s/it]]  
Number of matches 38074  
Number of matches After Lowe's Ratio 1360  
Number of Robust matches 752

48% | [29/61 [03:31<04:49, 9.05s/it]]  
Number of matches 33707  
Number of matches After Lowe's Ratio 1164  
Number of Robust matches 503

49% | [30/61 [03:38<04:21, 8.44s/it]]  
Number of matches 32337  
Number of matches After Lowe's Ratio 963  
Number of Robust matches 511

51% | [31/61 [03:44<03:58, 7.93s/it]]  
Number of matches 31634  
Number of matches After Lowe's Ratio 1097  
Number of Robust matches 555

52% | [32/61 [03:51<03:38, 7.54s/it]]  
Number of matches 31779  
Number of matches After Lowe's Ratio 1416  
Number of Robust matches 791

54% | [33/61 [03:58<03:25, 7.32s/it]]  
Number of matches 32533  
Number of matches After Lowe's Ratio 2572  
Number of Robust matches 1216

56% | [34/61 [04:05<03:14, 7.19s/it]]  
Number of matches 34689  
Number of matches After Lowe's Ratio 1301  
Number of Robust matches 690

57% | [35/61 [04:12<03:09, 7.29s/it]]  
Number of matches 32975  
Number of matches After Lowe's Ratio 2366  
Number of Robust matches 1620

59% | [36/61 [04:19<02:59, 7.19s/it]]  
Number of matches 32088  
Number of matches After Lowe's Ratio 1764  
Number of Robust matches 1231

61% | [37/61 [04:26<02:49, 7.05s/it]]  
Number of matches 32233  
Number of matches After Lowe's Ratio 1685  
Number of Robust matches 1282

62% | [38/61 [04:33<02:41, 7.04s/it]]  
Number of matches 30573  
Number of matches After Lowe's Ratio 1456  
Number of Robust matches 1383

64% | [39/61 [04:39<02:28, 6.77s/it]]  
Number of matches 30768  
Number of matches After Lowe's Ratio 1570  
Number of Robust matches 1377

66% | [40/61 [04:45<02:16, 6.52s/it]]  
Number of matches 26813  
Number of matches After Lowe's Ratio 1078  
Number of Robust matches 955

67% | [41/61 [04:50<02:02, 6.15s/it]]  
Number of matches 26443  
Number of matches After Lowe's Ratio 1948  
Number of Robust matches 1587

69% | 42/61 [04:56<01:52, 5.95s/it]

Number of matches 28580  
Number of matches After Lowe's Ratio 979  
Number of Robust matches 700

70% | 43/61 [05:02<01:46, 5.90s/it]

Number of matches 28143  
Number of matches After Lowe's Ratio 2266  
Number of Robust matches 1848

72% | 44/61 [05:07<01:35, 5.63s/it]

Number of matches 17084  
Number of matches After Lowe's Ratio 822  
Number of Robust matches 699

74% | 45/61 [05:09<01:16, 4.77s/it]

Number of matches 13636  
Number of matches After Lowe's Ratio 1343  
Number of Robust matches 1035

75% | 46/61 [05:12<01:00, 4.02s/it]

Number of matches 13604  
Number of matches After Lowe's Ratio 542  
Number of Robust matches 410

77% | 47/61 [05:14<00:48, 3.45s/it]

Number of matches 16334  
Number of matches After Lowe's Ratio 1470  
Number of Robust matches 1150

79% | 48/61 [05:17<00:42, 3.27s/it]

Number of matches 28245  
Number of matches After Lowe's Ratio 796  
Number of Robust matches 713

80% | 49/61 [05:23<00:49, 4.09s/it]

Number of matches 31687  
Number of matches After Lowe's Ratio 1596  
Number of Robust matches 1276

82% | 50/61 [05:30<00:55, 5.03s/it]

Number of matches 33859  
Number of matches After Lowe's Ratio 1401  
Number of Robust matches 1193

84% | 51/61 [05:37<00:56, 5.65s/it]

Number of matches 28095  
Number of matches After Lowe's Ratio 1291  
Number of Robust matches 1071

85% | 52/61 [05:43<00:50, 5.66s/it]

Number of matches 27389  
Number of matches After Lowe's Ratio 1435  
Number of Robust matches 1059

87% | 53/61 [05:48<00:45, 5.73s/it]

Number of matches 32426  
Number of matches After Lowe's Ratio 1002  
Number of Robust matches 833

89% | 54/61 [05:55<00:42, 6.05s/it]

Number of matches 30438  
Number of matches After Lowe's Ratio 241  
Number of Robust matches 175

90% | 55/61 [06:01<00:36, 6.08s/it]

Number of matches 28440  
Number of matches After Lowe's Ratio 883  
Number of Robust matches 730

92% | 56/61 [06:07<00:30, 6.02s/it]

Number of matches 26824  
Number of matches After Lowe's Ratio 1776  
Number of Robust matches 1250

93% | 57/61 [06:13<00:23, 5.82s/it]

Number of matches 26741  
Number of matches After Lowe's Ratio 1004  
Number of Robust matches 703

95% | 58/61 [06:18<00:16, 5.61s/it]

Number of matches 22819  
Number of matches After Lowe's Ratio 1769  
Number of Robust matches 1479

97% | 59/61 [06:22<00:10, 5.10s/it]

Number of matches 17913  
Number of matches After Lowe's Ratio 1985  
Number of Robust matches 1713

98% | 60/61 [06:25<00:04, 4.53s/it]

Number of matches 16676  
Number of matches After Lowe's Ratio 2017  
Number of Robust matches 1710

```
In [30]: 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 [31]: def final_steps_left(images_left,images_right,H_left,H_right,xmax,xmin,ymax,ymin,t,h,w,Ht):
    warp_imgs_left = []

    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))

        if j==0:
            result[t[1]:t[1]+t[1], t[0]:w+t[0]] = images_left[0]

        warp_imgs_left.append(result)

    print('Step31:Done')
    return warp_imgs_left

def final_steps_right(images_left,images_right,H_left,H_right,xmax,xmin,ymax,ymin,t,h,w,Ht):
    warp_imgs_right = []

    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_imgs_right.append(result)

    print('Step32:Done')
    return warp_imgs_right

def final_steps_union(warp_imgs_left,warp_imgs_right):
    #Union
    warp_images_all = warp_imgs_left + warp_imgs_right
    warp_img_init = warp_images_all[0]

    #warp_final_all=[]
    for j,warp_img in enumerate(warp_images_all):
        if j==len(warp_images_all)-1:
            break
        black_pixels = np.where((warp_img_init[:, :, 0] == 0) & (warp_img_init[:, :, 1] == 0) & (warp_img_init[:, :, 2] == 0))

        warp_img_init[black_pixels] = warp_images_all[j+1][black_pixels]

    #warp_final = np.maximum(warp_img_init,warp_images_all[j+1])
    #warp_img_init = warp_final
    #warp_final_all.append(warp_final)

    print('Step4:Done')

    return warp_img_init

```

```
In [32]: 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
    input_img = images_left[j+1]
    result = np.zeros((ymax-ymin,xmax-xmin,3),dtype='uint8')

    cv2.warpPerspective(src = np.uint8(input_img), M = H_trans, dsize = (xmax-xmin, ymax-ymin),dst=result)
    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_steps_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
    input_img = images_right[j+1]
    result = np.zeros((ymax-ymin,xmax-xmin,3),dtype='uint8')

    cv2.warpPerspective(src = np.uint8(input_img), M = H_trans, dsize = (xmax-xmin, ymax-ymin),dst=result)
    warp_img_init_curr = result

    black_pixels = np.where((warp_img_prev[:, :, 0] == 0) & (warp_img_prev[:, :, 1] == 0) & (warp_img_prev[:, :, 2] == 0))

    warp_img_prev[black_pixels] = warp_img_init_curr[black_pixels]

print('Step32:Done')

return warp_img_prev

```

In [33]: xmax,xmin,ymax,ymin,t,h,w,Ht = warpnImages(images\_left\_bgr\_no\_enhance, images\_right\_bgr\_no\_enhance,H\_left\_rootsift,H\_right\_rootsift)

Step1:Done  
Step2:Done

In [ ]: warp\_imgs\_left = final\_steps\_left\_union(images\_left\_bgr\_no\_enhance,H\_left\_rootsift,xmax,xmin,ymax,ymin,t,h,w,Ht)