In [2]:
```
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 [3]:
```
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 [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 /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 [4]:
```
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=a[j]
    listx=x.tolist()
    x.sort()
    minval1=x[0]                                #min
    minval2=x[1]                                #2nd min
    itemindex1 = listx.index(minval1)           #index of min val
    itemindex2 = listx.index(minval2)           #index of second min value
    ratio=minval1/minval2                       #Ratio Test

    if ratio<threshold:
      #Low distance ratio: fb1 can be a good match
      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_Homography(im1_pts,im2_pts):
  """
  im1_pts and im2_pts are 2×n 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^TA.
  #Columns of U are the eigenvectors of AA^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 [5]:
```
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 [9]:
```
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[55]
left_files_path_rev = []
right_files_path = []

for file in files_all[:56]:
  left_files_path_rev.append(folder_path + file)

left_files_path = left_files_path_rev[::-1]

for file in files_all[54:110]:
  right_files_path.append(folder_path + file)
```

In [10]:
```
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%|██████████| 56/56 [02:24<00:00,  2.59s/it]
100%|██████████| 56/56 [02:08<00:00,  2.30s/it]
```

In [11]:
```
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%|██████████| 56/56 [00:19<00:00,  2.82it/s]
100%|██████████| 56/56 [00:19<00:00,  2.87it/s]
```

In [36]:
```
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]))
```

```
100%|██████████| 56/56 [05:08<00:00,  5.51s/it]
100%|██████████| 56/56 [05:01<00:00,  5.38s/it]
```

In [37]:
```
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 [38]:
```
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 [39]:
```
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_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 [40]:
```
from functools import partial
from tqdm import tqdm
tqdm = partial(tqdm, position=0, leave=True)
```

In [41]:
```
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],0.5)
  H_left_surf.append(H_a)
  num_matches_surf.append(matches)
  num_good_matches_surf.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_surf[j:j+2][::-1],points_all_right_surf[j:j+2][::-1],descriptors_all_right_surf[j:j+2][::-1],0.5)
  H_right_surf.append(H_a)
```

```
  2%|▌          | 1/56 [00:03<03:24,  3.72s/it]
Number of matches 38139
Number of matches After Lowe's Ratio 610
Number of Robust matches 357


  4%|▉          | 2/56 [00:07<03:16,  3.64s/it]
Number of matches 39467
Number of matches After Lowe's Ratio 562
Number of Robust matches 404


  5%|▏         | 3/56 [00:10<03:06,  3.52s/it]
Number of matches 36010
Number of matches After Lowe's Ratio 326
Number of Robust matches 222


  7%|▍          | 4/56 [00:13<03:01,  3.49s/it]
Number of matches 38370
Number of matches After Lowe's Ratio 365
Number of Robust matches 278


  9%|▊          | 5/56 [00:16<02:51,  3.35s/it]
Number of matches 32669
Number of matches After Lowe's Ratio 241
Number of Robust matches 165


 11%|▉         | 6/56 [00:19<02:39,  3.19s/it]
Number of matches 34564
Number of matches After Lowe's Ratio 622
Number of Robust matches 515
```

 12%|▉ | 7/56 [00:22<02:31, 3.09s/it]
Number of matches 34700
Number of matches After Lowe's Ratio 443
Number of Robust matches 352

 14%|▉ | 8/56 [00:25<02:27, 3.07s/it]
Number of matches 37406
Number of matches After Lowe's Ratio 637
Number of Robust matches 489

 16%|▉ | 9/56 [00:29<02:30, 3.19s/it]
Number of matches 38907
Number of matches After Lowe's Ratio 1269
Number of Robust matches 1085

 18%|█▏ | 10/56 [00:32<02:29, 3.26s/it]
Number of matches 37650
Number of matches After Lowe's Ratio 1152
Number of Robust matches 947

 20%|█▏ | 11/56 [00:35<02:29, 3.33s/it]
Number of matches 38049
Number of matches After Lowe's Ratio 1191
Number of Robust matches 972

 21%|█▎ | 12/56 [00:39<02:28, 3.38s/it]
Number of matches 38270
Number of matches After Lowe's Ratio 1291
Number of Robust matches 1193

 23%|█▍ | 13/56 [00:42<02:25, 3.39s/it]
Number of matches 38222
Number of matches After Lowe's Ratio 2054
Number of Robust matches 1452

 25%|█▌ | 14/56 [00:46<02:23, 3.42s/it]
Number of matches 37766
Number of matches After Lowe's Ratio 2204
Number of Robust matches 1891

 27%|█▋ | 15/56 [00:49<02:20, 3.43s/it]
Number of matches 38678
Number of matches After Lowe's Ratio 1527
Number of Robust matches 1328

 29%|█▋ | 16/56 [00:53<02:19, 3.50s/it]
Number of matches 38137
Number of matches After Lowe's Ratio 892
Number of Robust matches 713

 30%|█▊ | 17/56 [00:56<02:16, 3.49s/it]
Number of matches 37525
Number of matches After Lowe's Ratio 1236
Number of Robust matches 1121

 32%|█▉ | 18/56 [01:00<02:11, 3.46s/it]
Number of matches 37759
Number of matches After Lowe's Ratio 648
Number of Robust matches 544

 34%|██ | 19/56 [01:03<02:09, 3.49s/it]
Number of matches 38066
Number of matches After Lowe's Ratio 791
Number of Robust matches 665

 36%|██▏ | 20/56 [01:07<02:05, 3.47s/it]
Number of matches 38808
Number of matches After Lowe's Ratio 225
Number of Robust matches 173

 38%|██▎ | 21/56 [01:10<02:02, 3.51s/it]
Number of matches 37519
Number of matches After Lowe's Ratio 262
Number of Robust matches 203

 39%|██▎ | 22/56 [01:14<01:57, 3.46s/it]
Number of matches 35706
Number of matches After Lowe's Ratio 599
Number of Robust matches 439

 41%|██▍ | 23/56 [01:17<01:55, 3.50s/it]
Number of matches 39467
Number of matches After Lowe's Ratio 321
Number of Robust matches 187

 43%|██▌ | 24/56 [01:21<01:52, 3.50s/it]
Number of matches 36038
Number of matches After Lowe's Ratio 156
Number of Robust matches 116

 45%|██▋ | 25/56 [01:24<01:47, 3.47s/it]
Number of matches 37306
Number of matches After Lowe's Ratio 376
Number of Robust matches 297

 46%|██▊ | 26/56 [01:28<01:46, 3.54s/it]
Number of matches 35977
Number of matches After Lowe's Ratio 162
Number of Robust matches 129

 48%|████▊ | 27/56 [01:31<01:40, 3.48s/it]
Number of matches 35736
Number of matches After Lowe's Ratio 41
Number of Robust matches 26

 50%|█████ | 28/56 [01:35<01:35, 3.42s/it]
Number of matches 36312
Number of matches After Lowe's Ratio 787
Number of Robust matches 564

 52%|█████▏ | 29/56 [01:38<01:32, 3.42s/it]
Number of matches 35698
Number of matches After Lowe's Ratio 817
Number of Robust matches 596

 54%|█████▎ | 30/56 [01:41<01:28, 3.40s/it]
Number of matches 36283
Number of matches After Lowe's Ratio 719
Number of Robust matches 628

 55%|█████▌ | 31/56 [01:45<01:26, 3.45s/it]
Number of matches 37204
Number of matches After Lowe's Ratio 738
Number of Robust matches 629

 57%|█████▋ | 32/56 [01:49<01:23, 3.50s/it]
Number of matches 38737
Number of matches After Lowe's Ratio 565
Number of Robust matches 362

 59%|█████▉ | 33/56 [01:52<01:20, 3.50s/it]
Number of matches 40144
Number of matches After Lowe's Ratio 656
Number of Robust matches 422

 61%|██████ | 34/56 [01:56<01:18, 3.56s/it]
Number of matches 39400
Number of matches After Lowe's Ratio 573
Number of Robust matches 455

 62%|██████▎ | 35/56 [01:59<01:14, 3.54s/it]
Number of matches 38368
Number of matches After Lowe's Ratio 886
Number of Robust matches 652

 64%|██████▍ | 36/56 [02:03<01:11, 3.55s/it]
Number of matches 38791
Number of matches After Lowe's Ratio 1242
Number of Robust matches 1076

 66%|██████▌ | 37/56 [02:06<01:06, 3.52s/it]
Number of matches 38100
Number of matches After Lowe's Ratio 1290
Number of Robust matches 983

 68%|██████▊ | 38/56 [02:10<01:03, 3.53s/it]
Number of matches 39195
Number of matches After Lowe's Ratio 1460
Number of Robust matches 1204

 70%|███████ | 39/56 [02:13<01:00, 3.53s/it]
Number of matches 40180
Number of matches After Lowe's Ratio 1425
Number of Robust matches 1216

 71%|███████▏ | 40/56 [02:17<00:56, 3.55s/it]
Number of matches 40454
Number of matches After Lowe's Ratio 1703
Number of Robust matches 1323

 73%|███████▎ | 41/56 [02:21<00:53, 3.58s/it]
Number of matches 39823
Number of matches After Lowe's Ratio 1736
Number of Robust matches 1269

 75%|███████▌ | 42/56 [02:24<00:49, 3.52s/it]
Number of matches 39851
Number of matches After Lowe's Ratio 1588
Number of Robust matches 1148

 77%|███████▋ | 43/56 [02:27<00:45, 3.51s/it]
Number of matches 37448
Number of matches After Lowe's Ratio 966
Number of Robust matches 840

 79%|███████▊ | 44/56 [02:31<00:41, 3.43s/it]
Number of matches 36805
Number of matches After Lowe's Ratio 2554
Number of Robust matches 2328

 80%|████████ | 45/56 [02:34<00:37, 3.40s/it]
Number of matches 39522
Number of matches After Lowe's Ratio 2129
Number of Robust matches 1807

 82%|████████▏ | 46/56 [02:37<00:33, 3.38s/it]
Number of matches 38125
Number of matches After Lowe's Ratio 1147

```
Number of Robust matches 950


 84%|▊▊▊▊▊▊▊▊  | 47/56 [02:41<00:30,  3.36s/it]
Number of matches 37529
Number of matches After Lowe's Ratio 1176
Number of Robust matches 963


 86%|▊▊▊▊▊▊▊▊  | 48/56 [02:44<00:27,  3.38s/it]
Number of matches 38691
Number of matches After Lowe's Ratio 1705
Number of Robust matches 1307


 88%|▊▊▊▊▊▊▊▊▊ | 49/56 [02:47<00:23,  3.34s/it]
Number of matches 35997
Number of matches After Lowe's Ratio 584
Number of Robust matches 417


 89%|▊▊▊▊▊▊▊▊▊ | 50/56 [02:51<00:19,  3.32s/it]
Number of matches 35242
Number of matches After Lowe's Ratio 691
Number of Robust matches 548


 91%|▊▊▊▊▊▊▊▊ | 51/56 [02:54<00:16,  3.23s/it]
Number of matches 36041
Number of matches After Lowe's Ratio 975
Number of Robust matches 671


 93%|▊▊▊▊▊▊▊▊▊ | 52/56 [02:57<00:12,  3.17s/it]
Number of matches 37450
Number of matches After Lowe's Ratio 1090
Number of Robust matches 672


 95%|▊▊▊▊▊▊▊▊▊ | 53/56 [03:00<00:09,  3.25s/it]
Number of matches 38454
Number of matches After Lowe's Ratio 525
Number of Robust matches 274


 96%|▊▊▊▊▊▊▊▊▊▊| 54/56 [03:03<00:06,  3.30s/it]
Number of matches 38388
Number of matches After Lowe's Ratio 895
Number of Robust matches 430


  0%|          | 0/56 [00:00<?, ?it/s]
Number of matches 38521
Number of matches After Lowe's Ratio 162
Number of Robust matches 91


  2%|▏         | 1/56 [00:03<03:12,  3.50s/it]
Number of matches 39256
Number of matches After Lowe's Ratio 676
Number of Robust matches 378


  4%|▎         | 2/56 [00:07<03:11,  3.55s/it]
Number of matches 42974
Number of matches After Lowe's Ratio 500
Number of Robust matches 305


  5%|▌         | 3/56 [00:11<03:13,  3.66s/it]
Number of matches 39642
Number of matches After Lowe's Ratio 823
Number of Robust matches 631


  7%|▋         | 4/56 [00:14<03:09,  3.64s/it]
Number of matches 41128
Number of matches After Lowe's Ratio 12
Number of Robust matches 8


  9%|▉         | 5/56 [00:18<03:08,  3.69s/it]
Number of matches 41423
Number of matches After Lowe's Ratio 118
Number of Robust matches 71


 11%|█         | 6/56 [00:22<03:05,  3.71s/it]
Number of matches 42807
Number of matches After Lowe's Ratio 266
Number of Robust matches 219


 12%|█▎        | 7/56 [00:26<03:04,  3.77s/it]
Number of matches 39765
Number of matches After Lowe's Ratio 536
Number of Robust matches 404


 14%|█▍        | 8/56 [00:29<02:57,  3.71s/it]
Number of matches 40727
Number of matches After Lowe's Ratio 992
Number of Robust matches 901


 16%|█▌        | 9/56 [00:33<02:50,  3.63s/it]
Number of matches 35726
Number of matches After Lowe's Ratio 1042
Number of Robust matches 866


 18%|█▊        | 10/56 [00:36<02:37,  3.42s/it]
Number of matches 29685
Number of matches After Lowe's Ratio 291
Number of Robust matches 228


 20%|█▉        | 11/56 [00:38<02:19,  3.09s/it]
Number of matches 32721
```

```
Number of matches After Lowe's Ratio 189
Number of Robust matches 167


 21%|██        | 12/56 [00:41<02:09,  2.95s/it]
Number of matches 29698
Number of matches After Lowe's Ratio 825
Number of Robust matches 595


 23%|██        | 13/56 [00:43<02:04,  2.89s/it]
Number of matches 39479
Number of matches After Lowe's Ratio 459
Number of Robust matches 416


 25%|██        | 14/56 [00:47<02:08,  3.05s/it]
Number of matches 37048
Number of matches After Lowe's Ratio 1352
Number of Robust matches 1035


 27%|██        | 15/56 [00:50<02:10,  3.19s/it]
Number of matches 39520
Number of matches After Lowe's Ratio 1798
Number of Robust matches 1532


 29%|██        | 16/56 [00:54<02:10,  3.25s/it]
Number of matches 36055
Number of matches After Lowe's Ratio 1744
Number of Robust matches 1377


 30%|███       | 17/56 [00:57<02:05,  3.21s/it]
Number of matches 36244
Number of matches After Lowe's Ratio 1773
Number of Robust matches 1523


 32%|███       | 18/56 [01:00<02:02,  3.23s/it]
Number of matches 35202
Number of matches After Lowe's Ratio 820
Number of Robust matches 731


 34%|███       | 19/56 [01:03<01:57,  3.16s/it]
Number of matches 37479
Number of matches After Lowe's Ratio 1179
Number of Robust matches 916


 36%|███       | 20/56 [01:06<01:57,  3.25s/it]
Number of matches 36769
Number of matches After Lowe's Ratio 880
Number of Robust matches 675


 38%|███       | 21/56 [01:10<01:54,  3.26s/it]
Number of matches 37985
Number of matches After Lowe's Ratio 814
Number of Robust matches 580


 39%|███       | 22/56 [01:13<01:51,  3.29s/it]
Number of matches 36813
Number of matches After Lowe's Ratio 790
Number of Robust matches 524


 41%|████      | 23/56 [01:16<01:48,  3.30s/it]
Number of matches 37749
Number of matches After Lowe's Ratio 1079
Number of Robust matches 667


 43%|████      | 24/56 [01:20<01:45,  3.28s/it]
Number of matches 39377
Number of matches After Lowe's Ratio 1014
Number of Robust matches 667


 45%|████      | 25/56 [01:23<01:44,  3.38s/it]
Number of matches 38479
Number of matches After Lowe's Ratio 1267
Number of Robust matches 761


 46%|████      | 26/56 [01:27<01:40,  3.36s/it]
Number of matches 35539
Number of matches After Lowe's Ratio 1221
Number of Robust matches 499


 48%|████      | 27/56 [01:30<01:35,  3.28s/it]
Number of matches 37695
Number of matches After Lowe's Ratio 738
Number of Robust matches 419


 50%|████      | 28/56 [01:33<01:33,  3.32s/it]
Number of matches 36113
Number of matches After Lowe's Ratio 514
Number of Robust matches 372


 52%|████      | 29/56 [01:36<01:29,  3.30s/it]
Number of matches 40129
Number of matches After Lowe's Ratio 75
Number of Robust matches 56


 54%|████      | 30/56 [01:40<01:29,  3.43s/it]
Number of matches 40100
Number of matches After Lowe's Ratio 202
Number of Robust matches 129
```

```
Number of matches 40735
Number of matches After Lowe's Ratio 2
Number of Robust matches 0
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-41-1a8471d2f8a8> in <module>()
     18     break
     19
---> 20   H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_surf[j:j+2][::-1],points_all_right_surf[j:j+2][::-1],descriptors_all_right_surf[j:j+2][::-1],
0.5)
     21   H_right_surf.append(H_a)

<ipython-input-39-c93470ca26fb> in get_Hmatrix(imgs, keypts, pts, descripts, ratio, thresh, disp)
     85     displayplot(dispimg1,'Robust Matching between Reference Image and Right Image ')
     86
---> 87   return Hn/Hn[2,2], len(matches_lf1_lf), len(inlier_matchset)

TypeError: 'NoneType' object is not subscriptable
```

In [45]:
```
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 [46]:
```
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]:h+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 [47]:
```
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 [ ]:
```
xmax,xmin,ymax,ymin,t,h,w,Ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_no_enhance,H_left_surf,H_right_surf)
```

In [ ]:
```
warp_imgs_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_surf,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

In [ ]:
```
warp_imgs_all_surf = final_steps_right_union(warp_imgs_left,images_right_bgr_no_enhance,H_right_surf,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

In [ ]:
```
fig,ax =plt.subplots()
fig.set_size_inches(20,20)
ax.imshow(cv2.cvtColor(warp_imgs_all_surf , cv2.COLOR_BGR2RGB))
ax.set_title('120-Images Mosaic-surf')
```

In [42]:
```
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 [44]:
```
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%|████████| 56/56 [02:09<00:00,  2.30s/it]
100%|████████| 56/56 [02:08<00:00,  2.30s/it]
```

In [49]:

```
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][::-
  H_right_rootsift.append(H_a)
```

```
  2%|▌          | 1/56 [00:04<04:25,  4.83s/it]
Number of matches 32266
Number of matches After Lowe's Ratio 4945
Number of Robust matches 2897


  4%|▌          | 2/56 [00:09<04:21,  4.84s/it]
Number of matches 32877
Number of matches After Lowe's Ratio 5446
Number of Robust matches 3000


  5%|▌         | 3/56 [00:14<04:11,  4.75s/it]
Number of matches 27613
Number of matches After Lowe's Ratio 2985
Number of Robust matches 1768


  7%|▊          | 4/56 [00:18<03:58,  4.59s/it]
Number of matches 31966
Number of matches After Lowe's Ratio 4286
Number of Robust matches 2763


  9%|▉          | 5/56 [00:22<03:47,  4.46s/it]
Number of matches 23666
Number of matches After Lowe's Ratio 1864
Number of Robust matches 1264


 11%|▉         | 6/56 [00:25<03:25,  4.11s/it]
Number of matches 27909
Number of matches After Lowe's Ratio 3926
Number of Robust matches 2541


 12%|▉          | 7/56 [00:29<03:16,  4.00s/it]
Number of matches 24162
Number of matches After Lowe's Ratio 3856
Number of Robust matches 2671


 14%|█          | 8/56 [00:33<03:02,  3.81s/it]
Number of matches 27592
Number of matches After Lowe's Ratio 4092
Number of Robust matches 2924


 16%|█         | 9/56 [00:36<02:57,  3.77s/it]
Number of matches 27282
Number of matches After Lowe's Ratio 5873
Number of Robust matches 4722


 18%|█▏        | 10/56 [00:40<02:55,  3.81s/it]
Number of matches 27099
Number of matches After Lowe's Ratio 4362
Number of Robust matches 3182


 20%|█▏        | 11/56 [00:44<02:48,  3.75s/it]
Number of matches 25919
Number of matches After Lowe's Ratio 4639
Number of Robust matches 3677


 21%|█▎        | 12/56 [00:47<02:41,  3.68s/it]
Number of matches 26581
Number of matches After Lowe's Ratio 4403
Number of Robust matches 3093


 23%|█▍        | 13/56 [00:51<02:38,  3.69s/it]
Number of matches 27052
Number of matches After Lowe's Ratio 5516
Number of Robust matches 4022


 25%|█▌       | 14/56 [00:55<02:35,  3.69s/it]
Number of matches 28182
Number of matches After Lowe's Ratio 6248
Number of Robust matches 4833


 27%|█▋        | 15/56 [00:59<02:36,  3.82s/it]
Number of matches 29913
```

```
Number of matches After Lowe's Ratio 5121
Number of Robust matches 3615


 29%|██▉       | 16/56 [01:03<02:40,  4.02s/it]
Number of matches 32208
Number of matches After Lowe's Ratio 4745
Number of Robust matches 3024


 30%|███       | 17/56 [01:08<02:44,  4.21s/it]
Number of matches 31209
Number of matches After Lowe's Ratio 4613
Number of Robust matches 2842


 32%|███▏      | 18/56 [01:13<02:45,  4.36s/it]
Number of matches 31666
Number of matches After Lowe's Ratio 5033
Number of Robust matches 3197


 34%|███▍      | 19/56 [01:17<02:43,  4.41s/it]
Number of matches 31039
Number of matches After Lowe's Ratio 4044
Number of Robust matches 2611


 36%|███▌      | 20/56 [01:22<02:43,  4.54s/it]
Number of matches 35893
Number of matches After Lowe's Ratio 3386
Number of Robust matches 1767


 38%|███▊      | 21/56 [01:27<02:46,  4.77s/it]
Number of matches 30633
Number of matches After Lowe's Ratio 3636
Number of Robust matches 2125


 39%|███▉      | 22/56 [01:32<02:41,  4.76s/it]
Number of matches 35656
Number of matches After Lowe's Ratio 3596
Number of Robust matches 1995


 41%|████      | 23/56 [01:38<02:45,  5.03s/it]
Number of matches 34271
Number of matches After Lowe's Ratio 4002
Number of Robust matches 1698


 43%|████▎     | 24/56 [01:43<02:46,  5.19s/it]
Number of matches 39115
Number of matches After Lowe's Ratio 2519
Number of Robust matches 1242


 45%|████▍     | 25/56 [01:50<02:54,  5.61s/it]
Number of matches 38049
Number of matches After Lowe's Ratio 4141
Number of Robust matches 2205


 46%|████▋     | 26/56 [01:56<02:54,  5.80s/it]
Number of matches 38187
Number of matches After Lowe's Ratio 2081
Number of Robust matches 1175


 48%|████▊     | 27/56 [02:02<02:49,  5.83s/it]
Number of matches 35025
Number of matches After Lowe's Ratio 920
Number of Robust matches 418


 50%|█████     | 28/56 [02:08<02:40,  5.74s/it]
Number of matches 35065
Number of matches After Lowe's Ratio 3714
Number of Robust matches 2212


 52%|█████▏    | 29/56 [02:13<02:31,  5.60s/it]
Number of matches 32733
Number of matches After Lowe's Ratio 4310
Number of Robust matches 2819


 54%|█████▎    | 30/56 [02:18<02:21,  5.44s/it]
Number of matches 33968
Number of matches After Lowe's Ratio 3775
Number of Robust matches 2452


 55%|█████▌    | 31/56 [02:23<02:15,  5.43s/it]
Number of matches 39070
Number of matches After Lowe's Ratio 4965
Number of Robust matches 2974


 57%|█████▋    | 32/56 [02:30<02:20,  5.84s/it]
Number of matches 41604
Number of matches After Lowe's Ratio 4663
Number of Robust matches 2102


 59%|█████▉    | 33/56 [02:37<02:24,  6.29s/it]
Number of matches 43881
Number of matches After Lowe's Ratio 4694
Number of Robust matches 1875


 61%|██████    | 34/56 [02:45<02:27,  6.69s/it]
Number of matches 40123
Number of matches After Lowe's Ratio 4324
Number of Robust matches 2058


 62%|██████▎   | 35/56 [02:52<02:20,  6.68s/it]
```

```
Number of matches 36596
Number of matches After Lowe's Ratio 4831
Number of Robust matches 2739


 64%|██████    | 36/56 [02:58<02:10,  6.50s/it]
Number of matches 33986
Number of matches After Lowe's Ratio 5114
Number of Robust matches 3139


 66%|██████    | 37/56 [03:03<01:55,  6.06s/it]
Number of matches 30652
Number of matches After Lowe's Ratio 4893
Number of Robust matches 3416


 68%|██████▏   | 38/56 [03:07<01:40,  5.60s/it]
Number of matches 29868
Number of matches After Lowe's Ratio 5577
Number of Robust matches 3740


 70%|██████▍   | 39/56 [03:12<01:29,  5.26s/it]
Number of matches 28588
Number of matches After Lowe's Ratio 4576
Number of Robust matches 3321


 71%|██████▍   | 40/56 [03:16<01:18,  4.88s/it]
Number of matches 28661
Number of matches After Lowe's Ratio 5147
Number of Robust matches 3385



Number of matches 27785
Number of matches After Lowe's Ratio 5697
 73%|██████▌   | 41/56 [03:20<01:09,  4.65s/it]
Number of Robust matches 3732


 75%|██████▊   | 42/56 [03:24<01:01,  4.40s/it]
Number of matches 27261
Number of matches After Lowe's Ratio 5447
Number of Robust matches 3426


 77%|██████▉   | 43/56 [03:27<00:53,  4.13s/it]
Number of matches 23943
Number of matches After Lowe's Ratio 3344
Number of Robust matches 2510


 79%|███████   | 44/56 [03:30<00:45,  3.82s/it]
Number of matches 23206
Number of matches After Lowe's Ratio 5619
Number of Robust matches 3988


 80%|███████▏  | 45/56 [03:33<00:39,  3.62s/it]
Number of matches 27133
Number of matches After Lowe's Ratio 5201
Number of Robust matches 4180


 82%|███████▍  | 46/56 [03:37<00:36,  3.67s/it]
Number of matches 28897
Number of matches After Lowe's Ratio 4114
Number of Robust matches 2921


 84%|███████▌  | 47/56 [03:41<00:34,  3.84s/it]
Number of matches 30356
Number of matches After Lowe's Ratio 4323
Number of Robust matches 2951


 86%|███████▋  | 48/56 [03:46<00:32,  4.01s/it]
Number of matches 30770
Number of matches After Lowe's Ratio 5567
Number of Robust matches 3764


 88%|███████▉  | 49/56 [03:50<00:28,  4.12s/it]
Number of matches 30376
Number of matches After Lowe's Ratio 4336
Number of Robust matches 2821


 89%|████████  | 50/56 [03:55<00:25,  4.27s/it]
Number of matches 30122
Number of matches After Lowe's Ratio 4734
Number of Robust matches 3103


 91%|████████▏ | 51/56 [03:59<00:21,  4.27s/it]
Number of matches 30579
Number of matches After Lowe's Ratio 4020
Number of Robust matches 2326


 93%|████████▍ | 52/56 [04:04<00:17,  4.30s/it]
Number of matches 29556
Number of matches After Lowe's Ratio 5810
Number of Robust matches 2929


 95%|████████▌ | 53/56 [04:08<00:13,  4.37s/it]
Number of matches 30860
Number of matches After Lowe's Ratio 3298
Number of Robust matches 1558


 96%|████████▋ | 54/56 [04:12<00:08,  4.39s/it]
Number of matches 30396
Number of matches After Lowe's Ratio 5318
Number of Robust matches 2977
```

```
  0%|          | 0/56 [00:00<?, ?it/s]
Number of matches 26655
Number of matches After Lowe's Ratio 1580
Number of Robust matches 642


  2%|▏         | 1/56 [00:04<04:34,  4.99s/it]
Number of matches 32463
Number of matches After Lowe's Ratio 5086
Number of Robust matches 2722


  4%|▎         | 2/56 [00:09<04:26,  4.94s/it]
Number of matches 32125
Number of matches After Lowe's Ratio 5240
Number of Robust matches 3285


  5%|▌         | 3/56 [00:14<04:23,  4.98s/it]
Number of matches 32332
Number of matches After Lowe's Ratio 4713
Number of Robust matches 2591


  7%|▋         | 4/56 [00:19<04:20,  5.01s/it]
Number of matches 35330
Number of matches After Lowe's Ratio 610
Number of Robust matches 231


  9%|▉         | 5/56 [00:25<04:17,  5.05s/it]
Number of matches 28871
Number of matches After Lowe's Ratio 1608
Number of Robust matches 760


 11%|█         | 6/56 [00:29<04:01,  4.83s/it]
Number of matches 30330
Number of matches After Lowe's Ratio 2319
Number of Robust matches 1359


 12%|█▎        | 7/56 [00:33<03:52,  4.74s/it]
Number of matches 32312
Number of matches After Lowe's Ratio 2406
Number of Robust matches 1378


 14%|█▍        | 8/56 [00:38<03:51,  4.83s/it]
Number of matches 32696
Number of matches After Lowe's Ratio 4012
Number of Robust matches 2777


 16%|█▌        | 9/56 [00:43<03:47,  4.83s/it]
Number of matches 31067
Number of matches After Lowe's Ratio 4213
Number of Robust matches 2775


 18%|█▊        | 10/56 [00:47<03:27,  4.52s/it]
Number of matches 19735
Number of matches After Lowe's Ratio 976
Number of Robust matches 679


 20%|█▉        | 11/56 [00:50<02:55,  3.91s/it]
Number of matches 27026
Number of matches After Lowe's Ratio 1121
Number of Robust matches 721


 21%|██        | 12/56 [00:53<02:47,  3.80s/it]
Number of matches 19200
Number of matches After Lowe's Ratio 2636
Number of Robust matches 2043


 23%|██▎       | 13/56 [00:56<02:26,  3.40s/it]
Number of matches 28248
Number of matches After Lowe's Ratio 1687
Number of Robust matches 1130


 25%|██▌       | 14/56 [01:00<02:30,  3.59s/it]
Number of matches 29428
Number of matches After Lowe's Ratio 5797
Number of Robust matches 4703


 27%|██▋       | 15/56 [01:04<02:36,  3.81s/it]
Number of matches 28737
Number of matches After Lowe's Ratio 5912
Number of Robust matches 4979


 29%|██▊       | 16/56 [01:08<02:35,  3.89s/it]
Number of matches 28036
Number of matches After Lowe's Ratio 4866
Number of Robust matches 4060


Number of matches 31014
Number of matches After Lowe's Ratio 4997
 30%|███       | 17/56 [01:12<02:36,  4.02s/it]
Number of Robust matches 4096


 32%|███▏      | 18/56 [01:17<02:40,  4.22s/it]
Number of matches 34946
Number of matches After Lowe's Ratio 3997
Number of Robust matches 2775


 34%|███▍      | 19/56 [01:23<02:50,  4.62s/it]
Number of matches 35939
```

```
Number of matches After Lowe's Ratio 5288
Number of Robust matches 3438


 36%|███       | 20/56 [01:28<02:59,  4.98s/it]
Number of matches 36837
Number of matches After Lowe's Ratio 4856
Number of Robust matches 3021


 38%|███▍      | 21/56 [01:34<03:03,  5.23s/it]
Number of matches 37856
Number of matches After Lowe's Ratio 5150
Number of Robust matches 2383


 39%|███▍      | 22/56 [01:40<03:07,  5.52s/it]
Number of matches 37941
Number of matches After Lowe's Ratio 5281
Number of Robust matches 2664


 41%|███▊      | 23/56 [01:46<03:04,  5.58s/it]
Number of matches 34580
Number of matches After Lowe's Ratio 5122
Number of Robust matches 2314


 43%|████      | 24/56 [01:52<02:56,  5.52s/it]
Number of matches 33372
Number of matches After Lowe's Ratio 4059
Number of Robust matches 2058


 45%|████      | 25/56 [01:57<02:47,  5.41s/it]
Number of matches 33599
Number of matches After Lowe's Ratio 5647
Number of Robust matches 2624



Number of matches 34878
Number of matches After Lowe's Ratio 4723
 46%|████▎     | 26/56 [02:02<02:42,  5.41s/it]
Number of Robust matches 2059


 48%|████▎     | 27/56 [02:08<02:37,  5.44s/it]
Number of matches 32992
Number of matches After Lowe's Ratio 4719
Number of Robust matches 2036


 50%|████▋     | 28/56 [02:13<02:29,  5.33s/it]
Number of matches 34056
Number of matches After Lowe's Ratio 3682
Number of Robust matches 1873


 52%|████▋     | 29/56 [02:19<02:28,  5.50s/it]
Number of matches 44723
Number of matches After Lowe's Ratio 1560
Number of Robust matches 726


 54%|█████     | 30/56 [02:26<02:41,  6.19s/it]
Number of matches 41727
Number of matches After Lowe's Ratio 3049
Number of Robust matches 1231


 55%|█████     | 31/56 [02:34<02:46,  6.67s/it]
Number of matches 46928
Number of matches After Lowe's Ratio 458
Number of Robust matches 158


 57%|█████▎    | 32/56 [02:42<02:48,  7.00s/it]
Number of matches 39889
Number of matches After Lowe's Ratio 2652
Number of Robust matches 1094


 59%|█████▋    | 33/56 [02:49<02:38,  6.91s/it]
Number of matches 38074
Number of matches After Lowe's Ratio 4928
Number of Robust matches 1982


 61%|█████▉    | 34/56 [02:55<02:25,  6.60s/it]
Number of matches 33707
Number of matches After Lowe's Ratio 3666
Number of Robust matches 1443


 62%|██████    | 35/56 [03:00<02:08,  6.13s/it]
Number of matches 32337
Number of matches After Lowe's Ratio 3253
Number of Robust matches 1364


 64%|██████    | 36/56 [03:04<01:54,  5.73s/it]
Number of matches 31634
Number of matches After Lowe's Ratio 3360
Number of Robust matches 1306


 66%|██████    | 37/56 [03:09<01:43,  5.43s/it]
Number of matches 31779
Number of matches After Lowe's Ratio 3844
Number of Robust matches 1554


 68%|██████▍   | 38/56 [03:14<01:35,  5.30s/it]
Number of matches 32533
Number of matches After Lowe's Ratio 6127
Number of Robust matches 2573
```

```
 70%|███████   | 39/56 [03:19<01:28,  5.21s/it]
Number of matches 34689
Number of matches After Lowe's Ratio 3711
Number of Robust matches 1520


 71%|███████   | 40/56 [03:24<01:24,  5.27s/it]
Number of matches 32975
Number of matches After Lowe's Ratio 5878
Number of Robust matches 3078


 73%|███████   | 41/56 [03:30<01:18,  5.23s/it]
Number of matches 32080
Number of matches After Lowe's Ratio 4526
Number of Robust matches 2391


 75%|███████   | 42/56 [03:35<01:12,  5.15s/it]
Number of matches 32233
Number of matches After Lowe's Ratio 3731
Number of Robust matches 2165


 77%|███████   | 43/56 [03:40<01:06,  5.10s/it]
Number of matches 30573
Number of matches After Lowe's Ratio 3253
Number of Robust matches 2554


 79%|███████   | 44/56 [03:44<00:59,  4.92s/it]
Number of matches 30768
Number of matches After Lowe's Ratio 3924
Number of Robust matches 2454


 80%|████████  | 45/56 [03:48<00:52,  4.75s/it]
Number of matches 26813
Number of matches After Lowe's Ratio 3001
Number of Robust matches 1962


 82%|████████  | 46/56 [03:52<00:44,  4.49s/it]
Number of matches 26443
Number of matches After Lowe's Ratio 4675
Number of Robust matches 3293


 84%|████████  | 47/56 [03:56<00:38,  4.25s/it]
Number of matches 28580
Number of matches After Lowe's Ratio 2755
Number of Robust matches 1409


 86%|████████  | 48/56 [04:00<00:33,  4.22s/it]
Number of matches 28143
Number of matches After Lowe's Ratio 5336
Number of Robust matches 3488


 88%|████████  | 49/56 [04:04<00:28,  4.02s/it]
Number of matches 17004
Number of matches After Lowe's Ratio 1411
Number of Robust matches 1128


 89%|████████  | 50/56 [04:05<00:20,  3.34s/it]
Number of matches 13636
Number of matches After Lowe's Ratio 2500
Number of Robust matches 1629


 91%|████████  | 51/56 [04:07<00:13,  2.74s/it]
Number of matches 13604
Number of matches After Lowe's Ratio 1406
Number of Robust matches 742


 93%|████████  | 52/56 [04:08<00:09,  2.35s/it]
Number of matches 16334
Number of matches After Lowe's Ratio 3305
Number of Robust matches 2339


 95%|████████  | 53/56 [04:10<00:06,  2.25s/it]
Number of matches 28245
Number of matches After Lowe's Ratio 1338
Number of Robust matches 986


 96%|████████  | 54/56 [04:15<00:05,  2.91s/it]
Number of matches 31687
Number of matches After Lowe's Ratio 4630
Number of Robust matches 3307


 98%|████████  | 55/56 [04:20<00:03,  3.51s/it]
Number of matches 33859
Number of matches After Lowe's Ratio 4299
Number of Robust matches 3047
```

In [50]:
```
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)
```