In [3]:

```
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
import h5py as h5

#cuda_output = !ldconfig -p|grep cudart.so|sed -e 's/.*\.\([0-9]*\)\.\([0-9]*\)$/cu\1\2/'
#accelerator = cuda_output[0] if exists('/dev/nvidia0') else 'cpu'

#print("Accelerator type = ",accelerator)
#print("Pytorch verision: ", torch.__version__)
```

In [1]:

```
from google.colab import drive
drive.mount('/content/drive')
```

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

In [ ]:

```
#!pip install ipython-autotime

#%load_ext autotime
```

In [2]:

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

In [ ]:

```
#!pip install opencv-python==4.4.0.44
#!pip install opencv-contrib-python==4.4.0.44
```

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 w
ith 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 es
timates
        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 [6]:

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

In [7]:

```
from zipfile import  ZipFile
file_name = '/content/drive/MyDrive/rgb-images.zip'

with ZipFile(file_name,'r') as zip:
  zip.extractall()
  print('Done')
```

```
Done
```

In [8]:

```
files_all=[]
for file in os.listdir("/content/RGB Images"):
    if file.endswith(".JPG"):
      files_all.append(file)


files_all.sort()
folder_path = '/content/RGB Images/'

#centre_file = folder_path + files_all[50]
left_files_path_rev = []
right_files_path = []
```

```
#Change this according to your dataset split

for file in files_all[:int(len(files_all)/2)+1]:
  left_files_path_rev.append(folder_path + file)

left_files_path = left_files_path_rev[::-1]

for file in files_all[int(len(files_all)/2):]:
  right_files_path.append(folder_path + file)
```

In [9]:

```
print(len(files_all))
```

```
113
```

In [10]:

```
from multiprocessing import Pool
```

In [11]:

```
#pool = Pool(4)

#images_left_bgr = pool.map(get_images, left_files_path)
```

In [12]:

```
import multiprocessing
print(multiprocessing.cpu_count())
```

```
4
```

In [13]:

```
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=0.75, fy=0.75, 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=0.75,fy=0.75, interpolation = cv2.INTER
_CUBIC )
  #images_right.append(cv2.cvtColor(right_img, cv2.COLOR_BGR2GRAY).astype('float32')/255.
)
  images_right_bgr.append(right_img)
```

```
100%|██████████| 57/57 [00:35<00:00,  1.60it/s]
100%|██████████| 57/57 [00:35<00:00,  1.60it/s]
```

In [14]:

```
Dataset = 'Industrial_Estate'
Dataset = 'Industrial_Estate'
```

In [15]:

```
f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','w')
t0=time.time()
f.create_dataset('data',data=images_left_bgr + images_right_bgr)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize(f'drive/MyDrive/a
ll_images_bgr_{Dataset}.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 70.14636945724487 [s] ... size 3840.164096 MB
```

In [16]:

```
f=h5.File(f'drive/MyDrive/all_images_gray_{Dataset}.h5','w')
t0=time.time()
f.create_dataset('data',data=images_left + images_right)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize(f'drive/MyDrive/a
ll_images_gray_{Dataset}.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.0037221908569335938 [s] ... size 0.0014 MB
```

In [17]:

```
del images_left_bgr,images_right_bgr
```

In [19]:

```
#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)
```

In [18]:

```
from timeit import default_timer as timer
```

In [19]:

```
time_all = []
```

In [20]:

```
num_kps_sift = []
num_kps_brisk = []
num_kps_agast = []
num_kps_kaze = []
num_kps_akaze = []
num_kps_orb = []
num_kps_mser = []
num_kps_daisy = []
num_kps_surfsift = []
num_kps_fast = []
num_kps_freak = []
num_kps_gftt = []
num_kps_briefstar = []
num_kps_surf = []
num_kps_rootsift = []
num_kps_superpoint = []
```

BRISK

In [20]:

```
Threshl=60;
Octaves=6;
#PatternScales=1.0f;

start = timer()

brisk = cv2.BRISK_create(Threshl,Octaves)


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

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

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = brisk.detect(imgs,None)
  kpt,descrip =  brisk.compute(imgs, kpt)
  keypoints_all_left_brisk.append(kpt)
  descriptors_all_left_brisk.append(descrip)
  #points_all_left_brisk.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = brisk.detect(imgs,None)
  kpt,descrip =  brisk.compute(imgs, kpt)
  keypoints_all_right_brisk.append(kpt)
  descriptors_all_right_brisk.append(descrip)
  #points_all_right_brisk.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [02:55<00:00,  3.07s/it]
100%|██████████| 57/57 [03:09<00:00,  3.33s/it]
```

In [21]:

```
for j in tqdm(keypoints_all_left_brisk + keypoints_all_right_brisk[1:]):
  num_kps_brisk.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 205087.13it/s]
```

In [22]:

```
all_feat_brisk_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_brisk):
  all_feat_brisk_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_brisk[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_brisk_left_each.append(temp)
  all_feat_brisk_left.append(all_feat_brisk_left_each)
```

In [23]:

```
all_feat_brisk_right = []
all_feat_brisk_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_brisk):
  all_feat_brisk_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_brisk[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_brisk_right_each.append(temp)
  all_feat_brisk_right.append(all_feat_brisk_right_each)
```

In [24]:

```
del keypoints_all_left_brisk, keypoints_all_right_brisk, descriptors_all_left_brisk, descriptors_all_right_brisk
```

In [25]:

```
import pickle
Fdb = open('all_feat_brisk_left.dat', 'wb')
pickle.dump(all_feat_brisk_left,Fdb,-1)
Fdb.close()
```

In [26]:

```
import pickle
Fdb = open('all_feat_brisk_right.dat', 'wb')
pickle.dump(all_feat_brisk_right,Fdb,-1)
Fdb.close()
```

In [27]:

```
del Fdb, all_feat_brisk_left, all_feat_brisk_right
```

ORB

In [28]:

```
orb = cv2.ORB_create(20000)

start = timer()


keypoints_all_left_orb = []
descriptors_all_left_orb = []
points_all_left_orb=[]

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

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = orb.detect(imgs,None)
  kpt,descrip =  orb.compute(imgs, kpt)
  keypoints_all_left_orb.append(kpt)
  descriptors_all_left_orb.append(descrip)
  #points_all_left_orb.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = orb.detect(imgs,None)
  kpt,descrip =  orb.compute(imgs, kpt)
  keypoints_all_right_orb.append(kpt)
  descriptors_all_right_orb.append(descrip)
  #points_all_right_orb.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

```
end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [00:38<00:00,  1.50it/s]
100%|██████████| 57/57 [00:39<00:00,  1.45it/s]
```

In [29]:

```
for j in tqdm(keypoints_all_left_orb + keypoints_all_right_orb[1:]):
  num_kps_orb.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 348497.32it/s]
```

In [30]:

```
all_feat_orb_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_orb):
  all_feat_orb_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_orb[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_orb_left_each.append(temp)
  all_feat_orb_left.append(all_feat_orb_left_each)
```

In [31]:

```
all_feat_orb_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_orb):
  all_feat_orb_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_orb[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_orb_right_each.append(temp)
  all_feat_orb_right.append(all_feat_orb_right_each)
```

In [32]:

```
del keypoints_all_left_orb, keypoints_all_right_orb, descriptors_all_left_orb, descriptor
s_all_right_orb
```

In [33]:

```
import pickle
Fdb = open('all_feat_orb_left.dat', 'wb')
pickle.dump(all_feat_orb_left,Fdb,-1)
Fdb.close()
```

In [34]:

```
import pickle
Fdb = open('all_feat_orb_right.dat', 'wb')
pickle.dump(all_feat_orb_right,Fdb,-1)
Fdb.close()
```

In [35]:

```
del Fdb, all_feat_orb_left, all_feat_orb_right
```

KAZE

In [1]:

```
start = timer()

kaze = cv2.KAZE_create()
```

```
keypoints_all_left_kaze = []
descriptors_all_left_kaze = []
points_all_left_kaze=[]

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

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = kaze.detect(imgs,None)
  kpt,descrip =  kaze.compute(imgs, kpt)
  keypoints_all_left_kaze.append(kpt)
  descriptors_all_left_kaze.append(descrip)
  #points_all_left_kaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = kaze.detect(imgs,None)
  kpt,descrip =  kaze.compute(imgs, kpt)
  keypoints_all_right_kaze.append(kpt)
  descriptors_all_right_kaze.append(descrip)
  #points_all_right_kaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-ba4386d34e72> in <module>()
----> 1 start = timer()
      2
      3 kaze = cv2.KAZE_create()
      4
      5

NameError: name 'timer' is not defined
```

In [ ]:

```
for j in tqdm(keypoints_all_left_kaze + keypoints_all_right_kaze[1:]):
  num_kps_kaze.append(len(j))
```

In [ ]:

```
all_feat_kaze_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_kaze):
  all_feat_kaze_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_kaze[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_kaze_left_each.append(temp)
  all_feat_kaze_left.append(all_feat_kaze_left_each)
```

In [ ]:

```
all_feat_kaze_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_kaze):
  all_feat_kaze_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_kaze[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_kaze_right_each.append(temp)
    all_feat_kaze_right.append(all_feat_kaze_right_each)
```

In [ ]:

```
del keypoints_all_left_kaze, keypoints_all_right_kaze, descriptors_all_left_kaze, descrip
tors_all_right_kaze
```

In [ ]:

```
import pickle
Fdb = open('all_feat_kaze_left.dat', 'wb')
pickle.dump(all_feat_kaze_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
import pickle
Fdb = open('all_feat_kaze_right.dat', 'wb')
pickle.dump(all_feat_kaze_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_kaze_left, all_feat_kaze_right
```

AKAZE

In [36]:

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

In [37]:

```
start = timer()

akaze = cv2.AKAZE_create()


keypoints_all_left_akaze = []
descriptors_all_left_akaze = []
points_all_left_akaze=[]

keypoints_all_right_akaze = []
descriptors_all_right_akaze = []
points_all_right_akaze=[]

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = akaze.detect(imgs,None)
  kpt,descrip =  akaze.compute(imgs, kpt)
  keypoints_all_left_akaze.append(kpt)
  descriptors_all_left_akaze.append(descrip)
  #points_all_left_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = akaze.detect(imgs,None)
  kpt,descrip = akaze.compute(imgs, kpt)
  keypoints_all_right_akaze.append(kpt)
  descriptors_all_right_akaze.append(descrip)
  #points_all_right_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()
```

```
time_all.append(end-start)
```

```
100%|██████████| 57/57 [03:42<00:00,  3.90s/it]
100%|██████████| 57/57 [03:48<00:00,  4.02s/it]
```

In [38]:

```
for j in tqdm(keypoints_all_left_akaze + keypoints_all_right_akaze[1:]):
  num_kps_akaze.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 92084.00it/s]
```

In [39]:

```
all_feat_akaze_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_akaze):
  all_feat_akaze_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_akaze[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_akaze_left_each.append(temp)
  all_feat_akaze_left.append(all_feat_akaze_left_each)
```

In [40]:

```
all_feat_akaze_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_akaze):
  all_feat_akaze_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_akaze[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_akaze_right_each.append(temp)
  all_feat_akaze_right.append(all_feat_akaze_right_each)
```

In [41]:

```
del keypoints_all_left_akaze, keypoints_all_right_akaze, descriptors_all_left_akaze, desc
riptors_all_right_akaze
```

In [42]:

```
import pickle
Fdb = open('all_feat_akaze_left.dat', 'wb')
pickle.dump(all_feat_akaze_left,Fdb,-1)
Fdb.close()
```

In [43]:

```
import pickle
Fdb = open('all_feat_akaze_right.dat', 'wb')
pickle.dump(all_feat_akaze_right,Fdb,-1)
Fdb.close()
```

In [44]:

```
del Fdb, all_feat_akaze_left, all_feat_akaze_right
```

STAR + BRIEF

In [46]:

```
start = timer()

star = cv2.xfeatures2d.StarDetector_create()
brief = cv2.xfeatures2d.BriefDescriptorExtractor_create()
```

```
keypoints_all_left_star = []
descriptors_all_left_brief = []
points_all_left_star=[]

keypoints_all_right_star = []
descriptors_all_right_brief = []
points_all_right_star=[]

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = star.detect(imgs,None)
  kpt,descrip =  brief.compute(imgs, kpt)
  keypoints_all_left_star.append(kpt)
  descriptors_all_left_brief.append(descrip)
  #points_all_left_star.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = star.detect(imgs,None)
  kpt,descrip =  brief.compute(imgs, kpt)
  keypoints_all_right_star.append(kpt)
  descriptors_all_right_brief.append(descrip)
  #points_all_right_star.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [02:56<00:00,  3.10s/it]
100%|██████████| 57/57 [02:49<00:00,  2.98s/it]
```

In [47]:

```
for j in tqdm(keypoints_all_left_star + keypoints_all_right_star[1:]):
  num_kps_briefstar.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 225157.41it/s]
```

In [48]:

```
all_feat_star_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_star):
  all_feat_star_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_brief[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_star_left_each.append(temp)
  all_feat_star_left.append(all_feat_star_left_each)
```

In [49]:

```
all_feat_star_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_star):
  all_feat_star_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_brief[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_star_right_each.append(temp)
  all_feat_star_right.append(all_feat_star_right_each)
```

In [50]:

```
del keypoints_all_left_star, keypoints_all_right_star, descriptors_all_left_brief, descri
ptors_all_right_brief
```

In [51]:

```
import pickle
Fdb = open('all_feat_star_left.dat', 'wb')
pickle.dump(all_feat_star_left,Fdb,-1)
Fdb.close()
```

In [52]:

```
import pickle
Fdb = open('all_feat_star_right.dat', 'wb')
pickle.dump(all_feat_star_right,Fdb,-1)
Fdb.close()
```

In [53]:

```
del Fdb, all_feat_star_left, all_feat_star_right
```

BRISK + FREAK

In [21]:

```
start = timer()

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

freak = cv2.xfeatures2d.FREAK_create()
keypoints_all_left_freak = []
descriptors_all_left_freak = []
points_all_left_freak=[]

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


for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = brisk.detect(imgs)
  kpt,descrip =  freak.compute(imgs, kpt)
  keypoints_all_left_freak.append(kpt)
  descriptors_all_left_freak.append(descrip)
  #points_all_left_freak.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = brisk.detect(imgs,None)
  kpt,descrip =  freak.compute(imgs, kpt)
  keypoints_all_right_freak.append(kpt)
  descriptors_all_right_freak.append(descrip)
  #points_all_right_freak.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [03:10<00:00,  3.34s/it]
100%|██████████| 57/57 [03:35<00:00,  3.77s/it]
```

In [22]:

```
for j in tqdm(keypoints_all_left_freak + keypoints_all_right_freak[1:]):
  num_kps_freak.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 100863.24it/s]
```

In [23]:

```
all_feat_freak_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_freak):
  all_feat_freak_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_freak[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_freak_left_each.append(temp)
  all_feat_freak_left.append(all_feat_freak_left_each)
```

In [24]:

```
all_feat_freak_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_freak):
  all_feat_freak_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_freak[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_freak_right_each.append(temp)
  all_feat_freak_right.append(all_feat_freak_right_each)
```

In [25]:

```
del keypoints_all_left_freak, keypoints_all_right_freak, descriptors_all_left_freak, desc
riptors_all_right_freak
```

In [26]:

```
import pickle
Fdb = open('all_feat_freak_left.dat', 'wb')
pickle.dump(all_feat_freak_left,Fdb,-1)
Fdb.close()
```

In [27]:

```
import pickle
Fdb = open('all_feat_freak_right.dat', 'wb')
pickle.dump(all_feat_freak_right,Fdb,-1)
Fdb.close()
```

In [28]:

```
del Fdb, all_feat_freak_left, all_feat_freak_right
```

MSER + SIFT

In [ ]:

```
start = timer()

mser = cv2.MSER_create()
sift = cv2.xfeatures2d.SIFT_create()

keypoints_all_left_mser = []
descriptors_all_left_mser = []
points_all_left_mser=[]

keypoints_all_right_mser = []
descriptors_all_right_mser = []
points_all_right_mser=[]

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = mser.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_left_mser.append(kpt)
  descriptors_all_left_mser.append(descrip)
  #points_all_left_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = mser.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_right_mser.append(kpt)
  descriptors_all_right_mser.append(descrip)
  #points_all_right_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

In [ ]:

```
for j in tqdm(keypoints_all_left_mser + keypoints_all_right_mser[1:]):
  num_kps_mser.append(len(j))
```

In [ ]:

```
all_feat_mser_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_mser):
  all_feat_mser_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_mser[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_mser_left_each.append(temp)
  all_feat_mser_left.append(all_feat_mser_left_each)
```

In [ ]:

```
all_feat_mser_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_mser):
  all_feat_mser_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_mser[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_mser_right_each.append(temp)
  all_feat_mser_right.append(all_feat_mser_right_each)
```

In [ ]:

```
del keypoints_all_left_mser, keypoints_all_right_mser, descriptors_all_left_mser, descrip
tors_all_right_mser
```

In [ ]:

```
import pickle
Fdb = open('all_feat_mser_left.dat', 'wb')
pickle.dump(all_feat_mser_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
import pickle
Fdb = open('all_feat_mser_right.dat', 'wb')
pickle.dump(all_feat_mser_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_mser_left, all_feat_mser_right
```

AGAST + SIFT

In [ ]:

```
start = timer()

agast = cv2.AgastFeatureDetector_create(threshold = 60)
sift = cv2.xfeatures2d.SIFT_create()

keypoints_all_left_agast = []
descriptors_all_left_agast = []
points_all_left_agast=[]

keypoints_all_right_agast = []
descriptors_all_right_agast = []
points_all_right_agast=[]

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = agast.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_left_agast.append(kpt)
  descriptors_all_left_agast.append(descrip)
  #points_all_left_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = agast.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_right_agast.append(kpt)
  descriptors_all_right_agast.append(descrip)
  #points_all_right_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [06:25<00:00,  6.76s/it]
100%|██████████| 57/57 [06:46<00:00,  7.13s/it]
```

In [ ]:

```
for j in tqdm(keypoints_all_left_agast + keypoints_all_right_agast[1:]):
  num_kps_agast.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 30997.80it/s]
```

In [ ]:

```
all_feat_agast_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_agast):
  all_feat_agast_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_agast[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_agast_left_each.append(temp)
  all_feat_agast_left.append(all_feat_agast_left_each)
```

In [ ]:

```
all_feat_agast_right = []
```

```
for cnt,kpt_all in enumerate(keypoints_all_right_agast):
  all_feat_agast_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_agast[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_agast_right_each.append(temp)
  all_feat_agast_right.append(all_feat_agast_right_each)
```

In [ ]:

```
del keypoints_all_left_agast, keypoints_all_right_agast, descriptors_all_left_agast, desc
riptors_all_right_agast
```

In [ ]:

```
import pickle
Fdb = open('all_feat_agast_left.dat', 'wb')
pickle.dump(all_feat_agast_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_agast_left
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-638aa3efa512> in <module>()
----> 1 del Fdb, all_feat_agast_left

NameError: name 'Fdb' is not defined
```

In [ ]:

```
import pickle
Fdb = open('all_feat_agast_right.dat', 'wb')
pickle.dump(all_feat_agast_right,Fdb,-1)
Fdb.close()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-129-700576f3a162> in <module>()
      1 import pickle
      2 Fdb = open('all_feat_agast_right.dat', 'wb')
----> 3 pickle.dump(all_feat_agast_right,Fdb,-1)
      4 Fdb.close()

NameError: name 'all_feat_agast_right' is not defined
```

In [ ]:

```
del Fdb, all_feat_agast_right
```

FAST + SIFT

In [ ]:

```
start = timer()


fast = cv2.FastFeatureDetector_create()
sift = cv2.xfeatures2d.SIFT_create()

keypoints_all_left_fast = []
descriptors_all_left_fast = []
points_all_left_fast=[]

keypoints_all_right_fast = []
descriptors_all_right_fast = []
```

```
points_all_right_fast=[]

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = fast.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_left_fast.append(kpt)
  descriptors_all_left_fast.append(descrip)
  #points_all_left_fast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = fast.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_right_fast.append(kpt)
  descriptors_all_right_fast.append(descrip)
  #points_all_right_fast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [03:31<00:00,  3.71s/it]
100%|██████████| 57/57 [03:32<00:00,  3.73s/it]
```

In [ ]:

```
for j in tqdm(keypoints_all_left_fast + keypoints_all_right_fast[1:]):
  num_kps_fast.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 33097.51it/s]
```

In [ ]:

```
all_feat_fast_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_fast):
  all_feat_fast_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_fast[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_fast_left_each.append(temp)
  all_feat_fast_left.append(all_feat_fast_left_each)
```

In [ ]:

```
all_feat_fast_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_fast):
  all_feat_fast_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_fast[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_fast_right_each.append(temp)
  all_feat_fast_right.append(all_feat_fast_right_each)
```

In [ ]:

```
del keypoints_all_left_fast, keypoints_all_right_fast, descriptors_all_left_fast, descrip
tors_all_right_fast
```

In [ ]:

```
import pickle
Fdb = open('all_feat_fast_left.dat', 'wb')
pickle.dump(all_feat_fast_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
import pickle
Fdb = open('all_feat_fast_right.dat', 'wb')
pickle.dump(all_feat_fast_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_fast_left, all_feat_fast_right
```

GFTT + SIFT

In [54]:

```
start = timer()

gftt = cv2.GFTTDetector_create()
sift = cv2.xfeatures2d.SIFT_create()

keypoints_all_left_gftt = []
descriptors_all_left_gftt = []
points_all_left_gftt=[]

keypoints_all_right_gftt = []
descriptors_all_right_gftt = []
points_all_right_gftt=[]

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = gftt.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_left_gftt.append(kpt)
  descriptors_all_left_gftt.append(descrip)
  #points_all_left_gftt.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = gftt.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_right_gftt.append(kpt)
  descriptors_all_right_gftt.append(descrip)
  #points_all_right_gftt.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [00:55<00:00,  1.03it/s]
100%|██████████| 57/57 [00:54<00:00,  1.04it/s]
```

In [55]:

```
for j in tqdm(keypoints_all_left_gftt + keypoints_all_right_gftt[1:]):
  num_kps_gftt.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 390408.86it/s]
```

In [56]:

```
all_feat_gftt_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_gftt):
  all_feat_gftt_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_gftt[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_gftt_left_each.append(temp)
  all_feat_gftt_left.append(all_feat_gftt_left_each)
```

In [57]:

```
all_feat_gftt_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_gftt):
  all_feat_gftt_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_gftt[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_gftt_right_each.append(temp)
  all_feat_gftt_right.append(all_feat_gftt_right_each)
```

In [58]:

```
del keypoints_all_left_gftt, keypoints_all_right_gftt, descriptors_all_left_gftt, descrip
tors_all_right_gftt
```

In [59]:

```
import pickle
Fdb = open('all_feat_gftt_left.dat', 'wb')
pickle.dump(all_feat_gftt_left,Fdb,-1)
Fdb.close()
```

In [60]:

```
import pickle
Fdb = open('all_feat_gftt_right.dat', 'wb')
pickle.dump(all_feat_gftt_right,Fdb,-1)
Fdb.close()
```

In [61]:

```
del Fdb, all_feat_gftt_left, all_feat_gftt_right
```

DAISY + SIFT

In [38]:

```
start = timer()

daisy = cv2.xfeatures2d.DAISY_create()
sift = cv2.xfeatures2d.SIFT_create()

keypoints_all_left_daisy = []
descriptors_all_left_daisy = []
points_all_left_daisy=[]

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

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = sift.detect(imgs,None)
  kpt,descrip =  daisy.compute(imgs, kpt)
  keypoints_all_left_daisy.append(kpt)
  descriptors_all_left_daisy.append(descrip)
  #points_all_left_daisy.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = sift.detect(imgs,None)
  kpt,descrip =  daisy.compute(imgs, kpt)
  keypoints_all_right_daisy.append(kpt)
  descriptors_all_right_daisy.append(descrip)
  #points_all_right_daisy.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [06:17<00:00,  6.62s/it]
100%|██████████| 57/57 [06:29<00:00,  6.84s/it]
```

In [39]:

```
for j in tqdm(keypoints_all_left_daisy + keypoints_all_right_daisy[1:]):
  num_kps_daisy.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 107375.70it/s]
```

In [40]:

```
all_feat_daisy_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_daisy):
  all_feat_daisy_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_daisy[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_daisy_left_each.append(temp)
  all_feat_daisy_left.append(all_feat_daisy_left_each)
```

In [ ]:

```
all_feat_daisy_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_daisy):
  all_feat_daisy_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_daisy[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_daisy_right_each.append(temp)
  all_feat_daisy_right.append(all_feat_daisy_right_each)
```

In [ ]:

```
del keypoints_all_left_daisy, keypoints_all_right_daisy, descriptors_all_left_daisy, desc
riptors_all_right_daisy
```

In [ ]:

```
import pickle
Fdb = open('all_feat_daisy_left.dat', 'wb')
pickle.dump(all_feat_daisy_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
import pickle
Fdb = open('all_feat_daisy_right.dat', 'wb')
pickle.dump(all_feat_daisy_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_daisy_left, all_feat_daisy_right
```

SURF + SIFT

SURF + SIFT

In [ ]:

```
'''
start = timer()

surf = cv2.xfeatures2d.SURF_create(upright=1)
sift = cv2.xfeatures2d.SIFT_create()

keypoints_all_left_surfsift = []
descriptors_all_left_surfsift = []
points_all_left_surfsift=[]

keypoints_all_right_surfsift = []
descriptors_all_right_surfsift = []
points_all_right_surfsift=[]

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  kpt = surf.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_left_surfsift.append(kpt)
  descriptors_all_left_surfsift.append(descrip)
  #points_all_left_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  kpt = surf.detect(imgs,None)
  kpt,descrip =  sift.compute(imgs, kpt)
  keypoints_all_right_surfsift.append(kpt)
  descriptors_all_right_surfsift.append(descrip)
  #points_all_right_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

end = timer()

time_all.append(end-start)
'''
```

Out[ ]:

```
"\nstart = timer()\n\nsurf = cv2.xfeatures2d.SURF_create(upright=1)\nsift = cv2.xfeatures
2d.SIFT_create()\n\nkeypoints_all_left_surfsift = []\ndescriptors_all_left_surfsift = []\
npoints_all_left_surfsift=[]\n\nkeypoints_all_right_surfsift = []\ndescriptors_all_right_
surfsift = []\npoints_all_right_surfsift=[]\n\nfor cnt in tqdm(range(len(left_files_path)
)):\n  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')\n  imgs = f['data'][cn
t]\n  f.close()     \n  kpt = surf.detect(imgs,None)\n  kpt,descrip =  sift.compute(imgs,
kpt)\n  keypoints_all_left_surfsift.append(kpt)\n  descriptors_all_left_surfsift.append(d
escrip)\n  #points_all_left_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt])
)\n\nfor cnt in tqdm(range(len(right_files_path))):\n  f=h5.File(f'drive/MyDrive/all_imag
es_bgr_{Dataset}.h5','r')\n  imgs = f['data'][cnt+len(left_files_path)]\n  f.close()  \n
kpt = surf.detect(imgs,None)\n  kpt,descrip =  sift.compute(imgs, kpt)\n  keypoints_all_r
ight_surfsift.append(kpt)\n  descriptors_all_right_surfsift.append(descrip)\n  #points_al
l_right_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))\n\nend = timer()\n
\ntime_all.append(end-start)\n"
```

In [ ]:

```
'''
for j in tqdm(keypoints_all_left_surfsift + keypoints_all_right_surfsift[1:]):
  num_kps_surfsift.append(len(j))
'''
```

In [ ]:

```
'''
all_feat_surfsift_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_surfsift):
  all_feat_surfsift_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_surfsift[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_surfsift_left_each.append(temp)
  all_feat_surfsift_left.append(all_feat_surfsift_left_each)
'''
```

In [ ]:

```
'''
all_feat_surfsift_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_surfsift):
  all_feat_surfsift_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_surfsift[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_surfsift_right_each.append(temp)
  all_feat_surfsift_right.append(all_feat_surfsift_right_each)
'''
```

In [ ]:

```
#del keypoints_all_left_surfsift, keypoints_all_right_surfsift, descriptors_all_left_surfsift, descriptors_all_right_surfsift
```

In [ ]:

```
'''
import pickle
Fdb = open('all_feat_surfsift_left.dat', 'wb')
pickle.dump(all_feat_surfsift_left,Fdb,-1)
Fdb.close()
'''
```

In [ ]:

```
'''
import pickle
Fdb = open('all_feat_surfsift_right.dat', 'wb')
pickle.dump(all_feat_surfsift_right,Fdb,-1)
Fdb.close()
'''
```

In [ ]:

```
#del Fdb, all_feat_surfsift_left, all_feat_surfsift_right
```

SIFT

In [ ]:

```
print(len(left_files_path))
```

```
57
```

In [ ]:

```
print(len(right_files_path))
```

In [ ]:

```
# H5 file w/o compression
#t0=time.time()
#f=h5.File('drive/MyDrive/all_images_bgr_sift.h5','r')
#print('HDF5  w/o comp.: data shape =',len(f['data'][0]),time.time()-t0,'[s]')
#f.close()
```

In [ ]:

```
#del f
```

In [117]:

```
start = timer()

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 cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  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 cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  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]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [09:22<00:00,  9.87s/it]
100%|██████████| 57/57 [10:10<00:00, 10.71s/it]
```

In [118]:

```
for j in tqdm(keypoints_all_left_sift + keypoints_all_right_sift[1:]):
  num_kps_sift.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 89341.44it/s]
```

In [119]:

```
all_feat_sift_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_sift):
  all_feat_sift_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_sift[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_sift_left_each.append(temp)
  all_feat_sift_left.append(all_feat_sift_left_each)
```

In [120]:

```
all_feat_sift_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_sift):
  all_feat_sift_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_sift[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_sift_right_each.append(temp)
  all_feat_sift_right.append(all_feat_sift_right_each)
```

In [121]:

```
del keypoints_all_left_sift, keypoints_all_right_sift, descriptors_all_left_sift, descrip
tors_all_right_sift
```

In [ ]:

```
import pickle
Fdb = open('all_feat_sift_left.dat', 'wb')
pickle.dump(all_feat_sift_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
import pickle
Fdb = open('all_feat_sift_right.dat', 'wb')
pickle.dump(all_feat_sift_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_sift_left, all_feat_sift_right
```

In [ ]:

```
#del keypoints_all_right_sift, keypoints_all_left_sift, descriptors_all_right_sift, descr
iptors_all_left_sift, points_all_right_sift, points_all_left_sift
```

SURF

In [ ]:

```
start = timer()

surf  = cv2.xfeatures2d.SURF_create(upright=1)
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 cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  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 cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  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]))
```

```
end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [01:53<00:00,  1.99s/it]
100%|██████████| 57/57 [01:52<00:00,  1.98s/it]
```

In [ ]:

```
for j in tqdm(keypoints_all_left_surf + keypoints_all_right_surf[1:]):
  num_kps_surf.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 544151.95it/s]
```

In [ ]:

```
all_feat_surf_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_surf):
  all_feat_surf_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_surf[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_surf_left_each.append(temp)
  all_feat_surf_left.append(all_feat_surf_left_each)
```

In [ ]:

```
all_feat_surf_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_surf):
  all_feat_surf_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_surf[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_surf_right_each.append(temp)
  all_feat_surf_right.append(all_feat_surf_right_each)
```

In [ ]:

```
del keypoints_all_left_surf, keypoints_all_right_surf, descriptors_all_left_surf, descrip
tors_all_right_surf
```

In [ ]:

```
import pickle
Fdb = open('all_feat_surf_left.dat', 'wb')
pickle.dump(all_feat_surf_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
import pickle
Fdb = open('all_feat_surf_right.dat', 'wb')
pickle.dump(all_feat_surf_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_surf_left, all_feat_surf_right
```

ROOTSIFT

In [29]:

```
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 [30]:

```
start = timer()

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 cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt]
  f.close()
  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 cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_bgr_{Dataset}.h5','r')
  imgs = f['data'][cnt+len(left_files_path)]
  f.close()
  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]))

end = timer()

time_all.append(end-start)
```

```
100%|██████████| 57/57 [08:55<00:00,  9.40s/it]
100%|██████████| 57/57 [09:27<00:00,  9.95s/it]
```

In [31]:

```
for j in tqdm(keypoints_all_left_rootsift + keypoints_all_right_rootsift[1:]):
  num_kps_rootsift.append(len(j))
```

```
100%|██████████| 113/113 [00:00<00:00, 150557.93it/s]
```

In [32]:

```
all_feat_rootsift_left = []
```

```
for cnt,kpt_all in enumerate(keypoints_all_left_rootsift):
  all_feat_rootsift_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_rootsift[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_rootsift_left_each.append(temp)
  all_feat_rootsift_left.append(all_feat_rootsift_left_each)
```

In [33]:

```
all_feat_rootsift_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_rootsift):
  all_feat_rootsift_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_rootsift[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_rootsift_right_each.append(temp)
  all_feat_rootsift_right.append(all_feat_rootsift_right_each)
```

In [36]:

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-36-2eaf24f94d4c> in <module>()
----> 1 del all_feat_rootsift_left, all_feat_rootsift_right, descriptors_all_left_rootsif
t, descriptors_all_right_rootsift

NameError: name 'descriptors_all_left_rootsift' is not defined
```

In [37]:

```
import pickle
Fdb = open('all_feat_rootsift_left.dat', 'wb')
pickle.dump(all_feat_rootsift_left,Fdb,-1)
Fdb.close()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-37-c8f2f6726f95> in <module>()
      1 import pickle
      2 Fdb = open('all_feat_rootsift_left.dat', 'wb')
----> 3 pickle.dump(all_feat_rootsift_left,Fdb,-1)
      4 Fdb.close()

NameError: name 'all_feat_rootsift_left' is not defined
```

In [ ]:

```
import pickle
Fdb = open('all_feat_rootsift_right.dat', 'wb')
pickle.dump(all_feat_rootsift_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_rootsift_left, all_feat_rootsift_right
```

SuperPoint

In [39]:

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

```
Cloning into 'SuperPointPretrainedNetwork'...
remote: Enumerating objects: 81, done.
remote: Total 81 (delta 0), reused 0 (delta 0), pack-reused 81
```

```
Unpacking objects: 100% (81/81), done.
```

In [40]:

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

cuda = False
```

In [41]:

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

In [42]:

```
import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F

torch.cuda.empty_cache()

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

    def forward(self, x):

        # Shared Encoder.
        x = self.relu(self.conv1a(x))
        x = self.relu(self.conv1b(x))
        x = self.pool(x)
        x = self.relu(self.conv2a(x))
        x = self.relu(self.conv2b(x))
        x = self.pool(x)
        x = self.relu(self.conv3a(x))
        x = self.relu(self.conv3b(x))
        x = self.pool(x)
        x = self.relu(self.conv4a(x))
        x = self.relu(self.conv4b(x))
        # Detector Head.
        cPa = self.relu(self.convPa(x))
        semi = self.convPb(cPa)
        # Descriptor Head.
        cDa = self.relu(self.convDa(x))
        desc = self.convDb(cDa)
        dn = torch.norm(desc, p=2, dim=1) # Compute the norm.
        desc = desc.div(torch.unsqueeze(dn, 1)) # Divide by norm to normalize.
        return semi, desc


class SuperPointFrontend(object):
    def __init__(self, weights_path, nms_dist, conf_thresh, nn_thresh,cuda=True):
        self.name = 'SuperPoint'
        self.cuda = cuda
        self.nms_dist = nms_dist
        self.conf_thresh = conf_thresh
        self.nn_thresh = nn_thresh # L2 descriptor distance for good match.
        self.cell = 8 # Size of each output cell. Keep this fixed.
        self.border_remove = 4 # Remove points this close to the border.

        # Load the network in inference mode.
        self.net = SuperPointNet()
        if cuda:
          # Train on GPU, deploy on GPU.
            self.net.load_state_dict(torch.load(weights_path))
            self.net = self.net.cuda()
        else:
          # Train on GPU, deploy on CPU.
            self.net.load_state_dict(torch.load(weights_path, map_location=lambda storage, loc: storage))
        self.net.eval()

    def nms_fast(self, in_corners, H, W, dist_thresh):

        grid = np.zeros((H, W)).astype(int) # Track NMS data.
        inds = np.zeros((H, W)).astype(int) # Store indices of points.
        # Sort by confidence and round to nearest int.
        inds1 = np.argsort(-in_corners[2,:])
        corners = in_corners[:,inds1]
        rcorners = corners[:2,:].round().astype(int) # Rounded corners.
        # Check for edge case of 0 or 1 corners.
        if rcorners.shape[1] == 0:
            return np.zeros((3,0)).astype(int), np.zeros(0).astype(int)
        if rcorners.shape[1] == 1:
            out = np.vstack((rcorners, in_corners[2])).reshape(3,1)
            return out, np.zeros((1)).astype(int)
        # Initialize the grid.
        for i, rc in enumerate(rcorners.T):
            grid[rcorners[1,i], rcorners[0,i]] = 1
            inds[rcorners[1,i], rcorners[0,i]] = i
        # Pad the border of the grid, so that we can NMS points near the border.
        pad = dist_thresh
        grid = np.pad(grid, ((pad,pad), (pad,pad)), mode='constant')
        # Iterate through points, highest to lowest conf, suppress neighborhood.
        count = 0
        for i, rc in enumerate(rcorners.T):
          # Account for top and left padding.
            pt = (rc[0]+pad, rc[1]+pad)
            if grid[pt[1], pt[0]] == 1: # If not yet suppressed.
                grid[pt[1]-pad:pt[1]+pad+1, pt[0]-pad:pt[0]+pad+1] = 0
                grid[pt[1], pt[0]] = -1
                count += 1
        # Get all surviving -1's and return sorted array of remaining corners.
        keepy, keepx = np.where(grid==-1)
        keepy, keepx = keepy - pad, keepx - pad
        inds_keep = inds[keepy, keepx]
        out = corners[:, inds_keep]
        values = out[-1, :]
        inds2 = np.argsort(-values)
        out = out[:, inds2]
        out_inds = inds1[inds_keep[inds2]]
        return out, out_inds

    def run(self, img):
        assert img.ndim == 2 #Image must be grayscale.
        assert img.dtype == np.float32 #Image must be float32.
        H, W = img.shape[0], img.shape[1]
        inp = img.copy()
        inp = (inp.reshape(1, H, W))
        inp = torch.from_numpy(inp)
        inp = torch.autograd.Variable(inp).view(1, 1, H, W)
        if self.cuda:
            inp = inp.cuda()
        # Forward pass of network.
        outs = self.net.forward(inp)
        semi, coarse_desc = outs[0], outs[1]
        # Convert pytorch -> numpy.
        semi = semi.data.cpu().numpy().squeeze()

        # --- Process points.
        dense = np.exp(semi) # Softmax.
        dense = dense / (np.sum(dense, axis=0)+.00001) # Should sum to 1.
        nodust = dense[:-1, :, :]
        # Reshape to get full resolution heatmap.
        Hc = int(H / self.cell)
        Wc = int(W / self.cell)
        nodust = np.transpose(nodust, [1, 2, 0])
        heatmap = np.reshape(nodust, [Hc, Wc, self.cell, self.cell])
        heatmap = np.transpose(heatmap, [0, 2, 1, 3])
        heatmap = np.reshape(heatmap, [Hc*self.cell, Wc*self.cell])
        prob_map = heatmap/np.sum(np.sum(heatmap))

        return heatmap, coarse_desc


    def key_pt_sampling(self, img, heat_map, coarse_desc, sampled):

        H, W = img.shape[0], img.shape[1]

        xs, ys = np.where(heat_map >= self.conf_thresh) # Confidence threshold.
        if len(xs) == 0:
            return np.zeros((3, 0)), None, None
        print("number of pts selected :", len(xs))


        pts = np.zeros((3, len(xs))) # Populate point data sized 3xN.
        pts[0, :] = ys
        pts[1, :] = xs
        pts[2, :] = heat_map[xs, ys]
        pts, _ = self.nms_fast(pts, H, W, dist_thresh=self.nms_dist) # Apply NMS.
        inds = np.argsort(pts[2,:])
        pts = pts[:,inds[::-1]] # Sort by confidence.
        bord = self.border_remove
        toremoveW = np.logical_or(pts[0, :] < bord, pts[0, :] >= (W-bord))
        toremoveH = np.logical_or(pts[1, :] < bord, pts[1, :] >= (H-bord))
        toremove = np.logical_or(toremoveW, toremoveH)
        pts = pts[:, ~toremove]
        pts = pts[:,0:sampled] #we take 2000 keypoints with highest probability from heat
map for our benchmark

        # --- Process descriptor.
        D = coarse_desc.shape[1]
        if pts.shape[1] == 0:
            desc = np.zeros((D, 0))
        else:
          # Interpolate into descriptor map using 2D point locations.
            samp_pts = torch.from_numpy(pts[:2, :].copy())
            samp_pts[0, :] = (samp_pts[0, :] / (float(W)/2.)) - 1.
            samp_pts[1, :] = (samp_pts[1, :] / (float(H)/2.)) - 1.
            samp_pts = samp_pts.transpose(0, 1).contiguous()
            samp_pts = samp_pts.view(1, 1, -1, 2)
            samp_pts = samp_pts.float()
            if self.cuda:
                samp_pts = samp_pts.cuda()
            desc = nn.functional.grid_sample(coarse_desc, samp_pts)
            desc = desc.data.cpu().numpy().reshape(D, -1)
            desc /= np.linalg.norm(desc, axis=0)[np.newaxis, :]


        return pts, desc
```

In [ ]:

```
print('Loading pre-trained network.')
# This class runs the SuperPoint network and processes its outputs.
```

```
fe = SuperPointFrontend(weights_path=weights_path,nms_dist = 3,conf_thresh = 0.01,nn_thr
esh=0.5)
print('Successfully loaded pre-trained network.')
```

In [ ]:

```
start = timer()

keypoints_all_left_superpoint = []
descriptors_all_left_superpoint = []
points_all_left_superpoint=[]

keypoints_all_right_superpoint = []
descriptors_all_right_superpoint = []
points_all_right_superpoint=[]

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

for cnt in tqdm(range(len(left_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_gray_{Dataset}.h5','r')
  lfpth = f['data'][cnt]
  f.close()
  heatmap1, coarse_desc1 = fe.run(lfpth)
  pts_1, desc_1 = fe.key_pt_sampling(lfpth, heatmap1, coarse_desc1, 80000) #Getting keyp
oints and descriptors for 1st image

  keypoints_all_left_superpoint.append(to_kpts(pts_1.T))
  descriptors_all_left_superpoint.append(desc_1.T)
  #points_all_left_superpoint.append(pts_1.T)


for cnt in tqdm(range(len(right_files_path))):
  f=h5.File(f'drive/MyDrive/all_images_gray_{Dataset}.h5','r')
  rfpth = f['data'][cnt]
  f.close()
  heatmap1, coarse_desc1 = fe.run(rfpth)
  pts_1, desc_1 = fe.key_pt_sampling(rfpth, heatmap1, coarse_desc1, 80000) #Getting keyp
oints and descriptors for 1st image

  keypoints_all_right_superpoint.append(to_kpts(pts_1.T))
  descriptors_all_right_superpoint.append(desc_1.T)
  #points_all_right_superpoint.append(pts_1.T)

end = timer()
time_all.append(end-start)
```

In [ ]:

```
for j in tqdm(keypoints_all_left_superpoint + keypoints_all_right_superpoint[1:]):
  num_kps_superpoint.append(len(j))
```

In [ ]:

```
all_feat_superpoint_left = []
for cnt,kpt_all in enumerate(keypoints_all_left_superpoint):
  all_feat_superpoint_left_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_left_superpoint[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_superpoint_left_each.append(temp)
  all_feat_superpoint_left.append(all_feat_superpoint_left_each)
```

In [ ]:

```
all_feat_superpoint_right = []
for cnt,kpt_all in enumerate(keypoints_all_right_superpoint):
  all_feat_superpoint_right_each = []
  for cnt_each, kpt in enumerate(kpt_all):
    desc = descriptors_all_right_superpoint[cnt][cnt_each]
    temp = (kpt.pt, kpt.size, kpt.angle, kpt.response, kpt.octave,
        kpt.class_id, desc)
    all_feat_superpoint_right_each.append(temp)
  all_feat_superpoint_right.append(all_feat_superpoint_right_each)
```

In [ ]:

```
del keypoints_all_left_superpoint, keypoints_all_right_superpoint, descriptors_all_left_s
uperpoint, descriptors_all_right_superpoint
```

In [ ]:

```
import pickle
Fdb = open('all_feat_superpoint_left.dat', 'wb')
pickle.dump(all_feat_superpoint_left,Fdb,-1)
Fdb.close()
```

In [ ]:

```
import pickle
Fdb = open('all_feat_superpoint_right.dat', 'wb')
pickle.dump(all_feat_superpoint_right,Fdb,-1)
Fdb.close()
```

In [ ]:

```
del Fdb, all_feat_superpoint_left, all_feat_superpoint_right
```

Total Matches,Robust Matches and Homography Computation

In [62]:

```
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, maxIters=
3000)
    inliers = inliers.flatten()
    return H, inliers
```

In [63]:

```
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 [64]:

```
def get_Hmatrix(imgs,keypts,pts,descripts,ratio=0.75,thresh=4,use_lowe=True,disp=False,no
_ransac=False,binary=False):
  lff1 = descripts[0]
  lff = descripts[1]

  if use_lowe==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()
    if binary==True:
      bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)

    else:
      bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
      lff1 = np.float32(descripts[0])
      lff = np.float32(descripts[1])


    #matches_lf1_lf = flann.knnMatch(lff1, lff, k=2)
    matches_4 = bf.knnMatch(lff1, lff,k=2)
    matches_lf1_lf = []


    print("\nNumber of matches",len(matches_4))
    '''
    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))
  else:
    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)
    if binary==True:
      bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
      lff1 = np.float32(descripts[0])
      lff = np.float32(descripts[1])
    else:
      bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
      lff1 = np.float32(descripts[0])
      lff = np.float32(descripts[1])


    matches_lf1_lf = flann.knnMatch(lff1, lff, k=2)
    #matches_lf1_lf = bf.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, RANSACthres
h=6)
  '''

  if no_ransac==True:
    Hn,inliers = compute_homography_fast_other(imm1_pts,imm2_pts)
  else:
    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)<25:
    matches_4 = []
    ratio = 0.85
    # 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(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, No
ne,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 [65]:

```
def get_Hmatrix_rfnet(imgs,pts,descripts,disp=True):

  des1 = descripts[0]
  des2 = descripts[1]

  kp1 = pts[0]
  kp2 = pts[1]


  predict_label, nn_kp2 = nearest_neighbor_distance_ratio_match(des1, des2, kp2, 0.7)
  idx = predict_label.nonzero().view(-1)
  mkp1 = kp1.index_select(dim=0, index=idx.long())  # predict match keypoints in I1
  mkp2 = nn_kp2.index_select(dim=0, index=idx.long())  # predict match keypoints in I2
  #img1, img2 = reverse_img(img1), reverse_img(img2)
  keypoints1 = list(map(to_cv2_kp, mkp1))
  keypoints2 = list(map(to_cv2_kp, mkp2))
  DMatch = list(map(to_cv2_dmatch, np.arange(0, len(keypoints1))))

  imm1_pts=np.empty((len(DMatch),2))
  imm2_pts=np.empty((len(DMatch),2))
  for i in range(0,len(DMatch)):
    m = DMatch[i]
    (a_x, a_y) = keypoints1[m.queryIdx].pt
    (b_x, b_y) = keypoints2[m.trainIdx].pt
    imm1_pts[i]=(a_x, a_y)
    imm2_pts[i]=(b_x, b_y)
  H=compute_Homography_fast(imm1_pts,imm2_pts)


  if disp==True:
    dispimg1 = cv2.drawMatches(imgs[0], keypoints1, imgs[1], keypoints2, DMatch, None)
    displayplot(dispimg1,'Robust Matching between Reference Image and Right Image ')


  return H/H[2,2]
```

In [66]:

```
import pickle
Fdb = open('all_feat_brisk_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

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

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_brisk.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_brisk.append(keypoints_each)
  descriptors_all_left_brisk.append(descrip_each)
```

In [67]:

```
import pickle
Fdb = open('all_feat_brisk_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

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

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_brisk.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each])
)
  keypoints_all_right_brisk.append(keypoints_each)
  descriptors_all_right_brisk.append(descrip_each)
```

In [68]:

```
H_left_brisk = []
H_right_brisk = []

num_matches_brisk = []
num_good_matches_brisk = []

images_left_bgr = []
images_right_bgr = []
for j in tqdm(range(len(left_files_path))):
  if j==len(left_files_path)-1:
    break

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_b
risk[j:j+2][::-1],points_all_left_brisk[j:j+2][::-1],descriptors_all_left_brisk[j:j+2][:
:-1],0.7,3,use_lowe=True,binary=True)
  H_left_brisk.append(H_a)
  num_matches_brisk.append(matches)
  num_good_matches_brisk.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_brisk[j:j+2][::-1],points_all_right_brisk[j:j+2][::-1],descriptors_all_right_brisk[j:j+
2][::-1],0.7,3,use_lowe=True,binary=True)
  H_right_brisk.append(H_a)
  num_matches_brisk.append(matches)
  num_good_matches_brisk.append(gd_matches)
```

```
  2%|▏         | 1/57 [00:02<02:35,  2.77s/it]
```

```
Number of matches 14597
Number of matches After Lowe's Ratio 187
Number of Robust matches 13


Number of matches After Lowe's Ratio New 1453
Number of Robust matches New 21
```

```
  4%|▎         | 2/57 [00:03<02:02,  2.23s/it]
```

```
Number of matches 22217
Number of matches After Lowe's Ratio 49
Number of Robust matches 14


Number of matches After Lowe's Ratio New 818
Number of Robust matches New 21
```

```
  5%|▌         | 3/57 [00:06<02:01,  2.25s/it]
```

```
Number of matches 47100
Number of matches After Lowe's Ratio 614
Number of Robust matches 166
```

```
  7%|▋         | 4/57 [00:53<13:58, 15.82s/it]
```

```
Number of matches 35278
Number of matches After Lowe's Ratio 229
Number of matches After Lowe's Ratio 229
Number of Robust matches 89
```

```
  9%|▉         | 5/57 [00:56<10:19, 11.92s/it]
```

```
Number of matches 27248
Number of matches After Lowe's Ratio 335
Number of Robust matches 157
```

```
 11%|█         | 6/57 [01:01<08:18,  9.77s/it]
```

```
Number of matches 70444
Number of matches After Lowe's Ratio 385
Number of Robust matches 184
```

```
 12%|█▏        | 7/57 [01:35<14:16, 17.13s/it]
```

```
Number of matches 260798
Number of matches After Lowe's Ratio 1163
Number of Robust matches 572
```

```
 14%|█▍        | 8/57 [02:12<18:57, 23.22s/it]
```

```
Number of matches 326014
Number of matches After Lowe's Ratio 1518
Number of Robust matches 546
```

```
 16%|█▌        | 9/57 [02:50<21:58, 27.46s/it]
```

```
Number of matches 337918
Number of matches After Lowe's Ratio 1155
Number of Robust matches 365
```

```
 18%|█▊        | 10/57 [03:36<25:53, 33.06s/it]
```

```
Number of matches 295074
Number of matches After Lowe's Ratio 1334
Number of Robust matches 357
```

```
 19%|█▉        | 11/57 [04:04<24:07, 31.47s/it]
```

```
Number of matches 180467
Number of matches After Lowe's Ratio 256
Number of Robust matches 84
```

```
 21%|██        | 12/57 [04:26<21:29, 28.66s/it]
```

```
Number of matches 173335
Number of matches After Lowe's Ratio 1253
Number of Robust matches 598
```

```
 23%|██▎       | 13/57 [04:47<19:18, 26.33s/it]
```

```
Number of matches 168090
Number of matches After Lowe's Ratio 673
Number of Robust matches 273
```

```
 25%|██▍       | 14/57 [04:56<15:14, 21.26s/it]
```

```
Number of matches 22134
Number of matches After Lowe's Ratio 45
Number of Robust matches 14


Number of matches After Lowe's Ratio New 861
Number of Robust matches New 29
```

```
 26%|██▋       | 15/57 [05:00<11:10, 15.97s/it]
```

```
Number of matches 60467
Number of matches After Lowe's Ratio 133
Number of Robust matches 27
```

```
 28%|██▊       | 16/57 [05:15<10:42, 15.67s/it]
```

```
Number of matches 147819
Number of matches After Lowe's Ratio 508
Number of Robust matches 278
```

```
 30%|██▉       | 17/57 [05:39<12:17, 18.43s/it]
```

```
Number of matches 274493
Number of matches After Lowe's Ratio 491
Number of Robust matches 327
```

```
 32%|███▏      | 18/57 [06:14<15:04, 23.20s/it]
```

```
Number of matches 324046
Number of matches After Lowe's Ratio 416
Number of Robust matches 119
```

```
 33%|███▎      | 19/57 [06:47<16:37, 26.26s/it]
```

```
Number of matches 301251
Number of matches After Lowe's Ratio 1090
Number of Robust matches 378
```

```
 35%|███▌      | 20/57 [07:13<16:08, 26.17s/it]
```

```
Number of matches 160272
Number of matches After Lowe's Ratio 444
Number of Robust matches 118
```

```
 37%|███▋      | 21/57 [07:24<12:57, 21.61s/it]
```

```
Number of matches 40395
Number of matches After Lowe's Ratio 843
Number of Robust matches 686
```

```
 39%|███▊      | 22/57 [07:27<09:18, 15.96s/it]
```

```
Number of matches 19141
Number of matches After Lowe's Ratio 293
Number of Robust matches 73
```

```
 40%|████      | 23/57 [07:29<06:36, 11.66s/it]
```

```
Number of matches 31128
Number of matches After Lowe's Ratio 73
Number of Robust matches 10


Number of matches After Lowe's Ratio New 1363
Number of Robust matches New 20
```

```
 42%|████▏     | 24/57 [07:33<05:10,  9.41s/it]
```

```
Number of matches 48223
Number of matches After Lowe's Ratio 293
Number of Robust matches 93
```

```
 44%|████▍     | 25/57 [07:40<04:45,  8.92s/it]
```

```
Number of matches 79786
Number of matches After Lowe's Ratio 295
Number of Robust matches 144
```

```
 46%|████▌     | 26/57 [07:47<04:14,  8.22s/it]
```

```
Number of matches 31100
Number of matches After Lowe's Ratio 248
Number of Robust matches 17


Number of matches After Lowe's Ratio New 2494
Number of Robust matches New 25
```

```
 47%|████▋     | 27/57 [07:49<03:13,  6.44s/it]
```

```
Number of matches 25292
Number of matches After Lowe's Ratio 126
Number of Robust matches 18


Number of matches After Lowe's Ratio New 1430
Number of Robust matches New 24
```

```
 49%|████▉     | 28/57 [07:51<02:27,  5.07s/it]
```

```
Number of matches 32538
Number of matches After Lowe's Ratio 352
Number of Robust matches 167
```

```
 51%|█████     | 29/57 [07:55<02:09,  4.63s/it]
```

```
Number of matches 27939
Number of matches After Lowe's Ratio 547
Number of Robust matches 179
```

```
 53%|█████▎    | 30/57 [07:57<01:46,  3.93s/it]
```

```
Number of matches 35608
Number of matches After Lowe's Ratio 244
Number of Robust matches 77
```

```
 54%|█████▍    | 31/57 [08:00<01:31,  3.54s/it]
```

```
Number of matches 23065
Number of matches After Lowe's Ratio 206
Number of Robust matches 54
```

```
 56%|█████▌    | 32/57 [08:03<01:22,  3.31s/it]
```

```
Number of matches 21668
Number of matches After Lowe's Ratio 229
Number of Robust matches 10


Number of matches After Lowe's Ratio New 2185
Number of Robust matches New 41
```

```
 58%|█████▊    | 33/57 [08:04<01:05,  2.74s/it]
```

```
Number of matches 19126
Number of matches After Lowe's Ratio 136
Number of Robust matches 34
```

```
 60%|█████▉    | 34/57 [08:06<00:55,  2.42s/it]
```

```
Number of matches 35425
Number of matches After Lowe's Ratio 349
Number of Robust matches 88
```

```
 61%|██████▏   | 35/57 [08:08<00:54,  2.46s/it]
```

```
Number of matches 20676
Number of matches After Lowe's Ratio 273
Number of Robust matches 62
```

```
 63%|██████▎   | 36/57 [08:11<00:53,  2.55s/it]
```

```
Number of matches 36243
Number of matches After Lowe's Ratio 182
Number of Robust matches 70
```

```
 65%|██████▍   | 37/57 [08:15<00:57,  2.89s/it]
```

```
Number of matches 44978
Number of matches After Lowe's Ratio 266
Number of Robust matches 187
```

```
 67%|██████▋   | 38/57 [08:25<01:35,  5.03s/it]
```

```
Number of matches 93842
Number of matches After Lowe's Ratio 304
Number of Robust matches 109
```

```
 68%|██████▊   | 39/57 [08:44<02:48,  9.37s/it]
```

```
Number of matches 149408
Number of matches After Lowe's Ratio 575
Number of Robust matches 303
```

```
 70%|███████   | 40/57 [09:00<03:12, 11.32s/it]
```

```
Number of matches 136590
Number of matches After Lowe's Ratio 321
Number of Robust matches 199
```

```
 72%|███████▏  | 41/57 [09:20<03:43, 13.99s/it]
```

```
Number of matches 180417
Number of matches After Lowe's Ratio 674
Number of Robust matches 366
```

```
 74%|███████▎  | 42/57 [09:33<03:23, 13.59s/it]
```

```
Number of matches 55988
Number of matches After Lowe's Ratio 439
Number of Robust matches 196
```

```
 75%|███████▌  | 43/57 [09:38<02:34, 11.05s/it]
```

```
Number of matches 36749
Number of matches After Lowe's Ratio 163
Number of Robust matches 94
```

```
 77%|███████▋  | 44/57 [09:44<02:03,  9.50s/it]
```

```
Number of matches 65924
Number of matches After Lowe's Ratio 203
Number of Robust matches 72
```

```
 79%|███████▉  | 45/57 [09:57<02:07, 10.60s/it]
```

```
Number of matches 120541
Number of matches After Lowe's Ratio 751
Number of Robust matches 313
```

```
 81%|████████  | 46/57 [10:17<02:26, 13.31s/it]
```

```
Number of matches 184409
Number of matches After Lowe's Ratio 279
Number of Robust matches 129
```

```
 82%|████████▏ | 47/57 [10:37<02:35, 15.52s/it]
```

```
Number of matches 173932
Number of matches After Lowe's Ratio 578
Number of Robust matches 238
```

```
 84%|████████▍ | 48/57 [10:52<02:17, 15.25s/it]
```

```
Number of matches 98237
Number of matches After Lowe's Ratio 1427
Number of Robust matches 651
```

```
 86%|████████▌ | 49/57 [11:00<01:45, 13.14s/it]
```

```
Number of matches 42253
Number of matches After Lowe's Ratio 119
Number of Robust matches 39
```

```
 88%|████████▊ | 50/57 [11:06<01:16, 10.86s/it]
```

```
Number of matches 62605
Number of matches After Lowe's Ratio 1032
Number of Robust matches 415
```

```
 89%|████████▉ | 51/57 [11:15<01:02, 10.47s/it]
```

```
Number of matches 74273
Number of matches After Lowe's Ratio 544
Number of Robust matches 318
```

```
 91%|█████████ | 52/57 [11:22<00:47,  9.40s/it]
```

```
Number of matches 37035
Number of matches After Lowe's Ratio 548
Number of Robust matches 256
```

```
 93%|█████████▎| 53/57 [11:25<00:29,  7.38s/it]
```

```
Number of matches 22525
Number of matches After Lowe's Ratio 392
Number of Robust matches 233
```

```
 95%|█████████▍| 54/57 [11:27<00:17,  5.72s/it]
```

```
Number of matches 39255
Number of matches After Lowe's Ratio 369
Number of Robust matches 131
```

```
 96%|█████████▋| 55/57 [11:30<00:09,  4.94s/it]
```

```
Number of matches 28544
Number of matches After Lowe's Ratio 354
Number of Robust matches 163
```

```
  0%|          | 0/57 [00:00<?, ?it/s]
```

```
Number of matches 59312
Number of matches After Lowe's Ratio 431
Number of Robust matches 260
```

```
  2%|▏         | 1/57 [00:03<03:07,  3.35s/it]
```

```
Number of matches 43746
Number of matches After Lowe's Ratio 621
Number of Robust matches 414
```

```
  4%|▎         | 2/57 [00:09<03:50,  4.19s/it]
```

```
Number of matches 64549
Number of matches After Lowe's Ratio 623
Number of Robust matches 310
```

```
  5%|▌         | 3/57 [00:16<04:29,  4.98s/it]
```

```
Number of matches 40287
Number of matches After Lowe's Ratio 919
Number of Robust matches 533
```

```
  7%|▋         | 4/57 [00:19<03:56,  4.46s/it]
```

```
Number of matches 26953
Number of matches After Lowe's Ratio 1466
Number of Robust matches 879
```

```
  9%|▉         | 5/57 [00:21<03:06,  3.58s/it]
```

```
Number of matches 12798
Number of matches After Lowe's Ratio 133
Number of Robust matches 39
```

```
 11%|█         | 6/57 [00:21<02:18,  2.72s/it]
```

```
Number of matches 9833
Number of matches After Lowe's Ratio 120
Number of Robust matches 19


Number of matches After Lowe's Ratio New 657
Number of Robust matches New 57
```

```
 12%|█▏        | 7/57 [00:23<02:05,  2.52s/it]
```

```
Number of matches 26347
Number of matches After Lowe's Ratio 84
Number of Robust matches 13


Number of matches After Lowe's Ratio New 1188
Number of Robust matches New 21
```

```
 14%|█▍        | 8/57 [00:26<02:01,  2.47s/it]
```

```
Number of matches 41375
Number of matches After Lowe's Ratio 267
Number of Robust matches 142
```

```
 16%|█▌        | 9/57 [00:29<02:11,  2.74s/it]
```

```
Number of matches 24778
Number of matches After Lowe's Ratio 104
Number of Robust matches 59
```

```
 18%|█▊        | 10/57 [00:33<02:28,  3.15s/it]
```

```
Number of matches 64571
Number of matches After Lowe's Ratio 445
Number of Robust matches 343
```

```
 19%|█▉        | 11/57 [00:42<03:36,  4.70s/it]
 19%|█▉        | 11/57 [00:42<03:36,  4.70s/it]
```

```
Number of matches 58304
Number of matches After Lowe's Ratio 227
Number of Robust matches 88
```

```
 21%|██        | 12/57 [01:04<07:30, 10.01s/it]
```

```
Number of matches 243060
Number of matches After Lowe's Ratio 402
Number of Robust matches 166
```

```
 23%|██▎       | 13/57 [01:27<10:16, 14.02s/it]
```

```
Number of matches 185393
Number of matches After Lowe's Ratio 729
Number of Robust matches 372
```

```
 25%|██▍       | 14/57 [01:56<13:13, 18.45s/it]
```

```
Number of matches 304462
Number of matches After Lowe's Ratio 659
Number of Robust matches 311
```

```
Number of matches 91711
Number of matches After Lowe's Ratio 45
Number of Robust matches 6
```

```
Number of matches After Lowe's Ratio New 3053
```

```
 26%|██▋       | 15/57 [02:22<14:24, 20.59s/it]
```

```
Number of Robust matches New 6
```

```
 28%|██▊       | 16/57 [02:49<15:32, 22.74s/it]
```

```
Number of matches 288940
Number of matches After Lowe's Ratio 1459
Number of Robust matches 603
```

```
 30%|██▉       | 17/57 [03:20<16:46, 25.16s/it]
```

```
Number of matches 260283
Number of matches After Lowe's Ratio 707
Number of Robust matches 307
```

```
Number of matches 346969
Number of matches After Lowe's Ratio 1092
Number of Robust matches 243
```

```
 33%|███▎      | 19/57 [04:36<20:08, 31.79s/it]
```

```
Number of matches 322855
Number of matches After Lowe's Ratio 805
Number of Robust matches 316
```

```
 35%|███▌      | 20/57 [05:11<20:15, 32.84s/it]
 35%|███▌      | 20/57 [05:11<20:15, 32.04s/it]
```

```
Number of matches 273075
Number of matches After Lowe's Ratio 2006
Number of Robust matches 656
```

```
 37%|███▋      | 21/57 [05:43<19:33, 32.60s/it]
```

```
Number of matches 261134
Number of matches After Lowe's Ratio 896
Number of Robust matches 355
```

```
 39%|███▊      | 22/57 [06:12<18:19, 31.43s/it]
```

```
Number of matches 244626
Number of matches After Lowe's Ratio 1371
Number of Robust matches 563
```

```
 40%|████      | 23/57 [06:33<15:57, 28.17s/it]
```

```
Number of matches 131035
Number of matches After Lowe's Ratio 980
Number of Robust matches 540
```

```
 42%|████▏     | 24/57 [06:44<12:44, 23.17s/it]
```

```
Number of matches 75551
Number of matches After Lowe's Ratio 1611
Number of Robust matches 725
```

```
 44%|████▍     | 25/57 [06:48<09:19, 17.48s/it]
```

```
Number of matches 10840
Number of matches After Lowe's Ratio 23
Number of Robust matches 10


Number of matches After Lowe's Ratio New 431
Number of Robust matches New 10
```

```
 46%|████▌     | 26/57 [06:49<06:27, 12.50s/it]
```

```
Number of matches 26999
Number of matches After Lowe's Ratio 73
Number of Robust matches 18


Number of matches After Lowe's Ratio New 1042
Number of Robust matches New 21
```

```
 47%|████▋     | 27/57 [06:51<04:41,  9.37s/it]
```

```
Number of matches 29869
Number of matches After Lowe's Ratio 195
Number of Robust matches 78
```

```
 49%|████▉     | 28/57 [06:54<03:39,  7.55s/it]
```

```
Number of matches 25454
Number of matches After Lowe's Ratio 137
Number of Robust matches 51
```

```
 51%|█████     | 29/57 [06:56<02:40,  5.73s/it]
```

```
Number of matches 15462
Number of matches After Lowe's Ratio 78
Number of Robust matches 28
```

```
 53%|█████▎    | 30/57 [06:57<01:54,  4.26s/it]
```

```
Number of matches 15467
Number of matches After Lowe's Ratio 104
Number of Robust matches 35
```

```
 54%|█████▍    | 31/57 [06:58<01:24,  3.27s/it]
```

```
Number of matches 22902
Number of matches After Lowe's Ratio 138
Number of Robust matches 37
```

```
 56%|█████▌    | 32/57 [07:02<01:26,  3.47s/it]
```

```
Number of matches 70002
Number of matches After Lowe's Ratio 1521
Number of Robust matches 1028
```

```
 58%|█████▊    | 33/57 [07:13<02:18,  5.78s/it]
```

```
Number of matches 94081
Number of matches After Lowe's Ratio 5702
Number of Robust matches 4188
```

```
 60%|█████▉    | 34/57 [07:20<02:20,  6.11s/it]
```

```
Number of matches 38421
Number of matches After Lowe's Ratio 1363
Number of Robust matches 888
```

```
 61%|██████▏   | 35/57 [07:27<02:20,  6.38s/it]
```

```
Number of matches 73113
Number of matches After Lowe's Ratio 435
Number of Robust matches 217
```

```
 63%|██████▎   | 36/57 [07:37<02:39,  7.57s/it]
```

```
Number of matches 80727
Number of matches After Lowe's Ratio 314
Number of Robust matches 133
```

```
 65%|██████▍   | 37/57 [07:49<02:58,  8.94s/it]
```

```
Number of matches 108828
Number of matches After Lowe's Ratio 53
Number of Robust matches 30
```

```
 67%|██████▋   | 38/57 [07:58<02:50,  9.00s/it]

Number of matches 47975
Number of matches After Lowe's Ratio 115
Number of Robust matches 83


 68%|██████▊   | 39/57 [08:04<02:25,  8.07s/it]

Number of matches 54837
Number of matches After Lowe's Ratio 118
Number of Robust matches 67


 70%|███████   | 40/57 [08:07<01:52,  6.60s/it]

Number of matches 9614
Number of matches After Lowe's Ratio 120
Number of Robust matches 63


 72%|███████▏  | 41/57 [08:10<01:24,  5.27s/it]

Number of matches 47513
Number of matches After Lowe's Ratio 306
Number of Robust matches 197


 74%|███████▎  | 42/57 [08:17<01:28,  5.88s/it]

Number of matches 57561
Number of matches After Lowe's Ratio 287
Number of Robust matches 190


 75%|███████▌  | 43/57 [08:25<01:30,  6.43s/it]

Number of matches 61811
Number of matches After Lowe's Ratio 253
Number of Robust matches 143


 77%|███████▋  | 44/57 [08:38<01:49,  8.43s/it]

Number of matches 117540
Number of matches After Lowe's Ratio 696
Number of Robust matches 333



Number of matches 190138
Number of matches After Lowe's Ratio 1715
 79%|███████▉  | 45/57 [09:00<02:30, 12.54s/it]

Number of Robust matches 894


 81%|████████  | 46/57 [09:25<03:01, 16.46s/it]

Number of matches 231727
Number of matches After Lowe's Ratio 501
Number of Robust matches 266



Number of matches 345749
Number of matches After Lowe's Ratio 1319
Number of Robust matches 485
```

```
 84%|████████▍ | 48/57 [10:44<04:09, 27.72s/it]
```

```
Number of matches 178093
Number of matches After Lowe's Ratio 690
Number of Robust matches 207
```

```
 86%|████████▌ | 49/57 [11:05<03:26, 25.85s/it]
```

```
Number of matches 136802
Number of matches After Lowe's Ratio 912
Number of Robust matches 564
```

```
 88%|████████▊ | 50/57 [11:21<02:38, 22.71s/it]
```

```
Number of matches 130711
Number of matches After Lowe's Ratio 1137
Number of Robust matches 582
```

```
 89%|████████▉ | 51/57 [11:36<02:01, 20.33s/it]
```

```
Number of matches 124537
Number of matches After Lowe's Ratio 1437
Number of Robust matches 796
```

```
 91%|█████████ | 52/57 [11:45<01:25, 17.11s/it]
```

```
Number of matches 43914
Number of matches After Lowe's Ratio 1153
Number of Robust matches 673
```

```
 93%|█████████▎| 53/57 [11:52<00:56, 14.00s/it]
```

```
Number of matches 58299
Number of matches After Lowe's Ratio 221
Number of Robust matches 92
```

```
 95%|█████████▍| 54/57 [12:00<00:36, 12.09s/it]
```

```
Number of matches 60153
Number of matches After Lowe's Ratio 1938
Number of Robust matches 1183
```

```
 96%|█████████▋| 55/57 [12:04<00:19,  9.70s/it]
```

```
Number of matches 17562
Number of matches After Lowe's Ratio 385
Number of Robust matches 278
```

```
 98%|█████████▊| 56/57 [12:08<00:08,  8.00s/it]
```

```
Number of matches 98226
Number of matches After Lowe's Ratio 201
Number of Robust matches 116
```

In [69]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_brisk_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_brisk)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_brisk_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.005534887313842773 4 [s] ... size 0.00608 MB
```

In [70]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_brisk_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_brisk)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_r
ight_brisk_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.0030717849731445312 [s] ... size 0.00608 MB
```

In [71]:

```
del H_left_brisk, H_right_brisk,keypoints_all_left_brisk, keypoints_all_right_brisk, desc
riptors_all_left_brisk, descriptors_all_right_brisk, points_all_left_brisk, points_all_ri
ght_brisk
```

In [ ]:

```
import pickle
Fdb = open('all_feat_sift_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_sift = []
descriptors_all_left_sift = []



for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_sift.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_sift.append(keypoints_each)
  descriptors_all_left_sift.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_sift_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_sift = []
descriptors_all_right_sift = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                                _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_sift.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_sift.append(keypoints_each)
  descriptors_all_right_sift.append(descrip_each)
```

In [ ]:

```
H_left_sift = []
H_right_sift = []

num_matches_sift = []
num_good_matches_sift = []

for j in tqdm(range(len(left_files_path))):
  if j==len(left_files_path)-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],0.75)
  H_left_sift.append(H_a)
  num_matches_sift.append(matches)
  num_good_matches_sift.append(gd_matches)

for j in tqdm(range(len(right_files_path))):
  if j==len(right_files_path)-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],0.75)
  H_right_sift.append(H_a)
  num_matches_sift.append(matches)
  num_good_matches_sift.append(gd_matches)
```

```
  2%|▏         | 1/57 [00:01<01:14,  1.33s/it]
```

```
Number of matches 8142
Number of matches After Lowe's Ratio 563
Number of Robust matches 219
```

```
  4%|▎         | 2/57 [00:02<01:09,  1.26s/it]
```

```
Number of matches 10780
Number of matches After Lowe's Ratio 483
Number of Robust matches 237
```

```
  5%|▌         | 3/57 [00:03<01:11,  1.32s/it]
```

```
Number of matches 14660
Number of matches After Lowe's Ratio 849
Number of Robust matches 128
```

```
  7%|▋         | 4/57 [00:05<01:19,  1.50s/it]
```

```
Number of matches 12881
Number of matches After Lowe's Ratio 378
Number of Robust matches 65
```

```
  9%|▉         | 5/57 [00:07<01:25,  1.64s/it]
  9%|▉         | 5/57 [00:07<01:25,  1.64s/it]
```

```
Number of matches 12861
Number of matches After Lowe's Ratio 967
Number of Robust matches 376
```

```
 11%|█         | 6/57 [00:09<01:26,  1.69s/it]
```

```
Number of matches 23189
Number of matches After Lowe's Ratio 696
Number of Robust matches 331
```

```
 12%|█▏        | 7/57 [00:14<02:14,  2.69s/it]
```

```
Number of matches 47705
Number of matches After Lowe's Ratio 2048
Number of Robust matches 1414
```

```
 14%|█▍        | 8/57 [00:26<04:26,  5.43s/it]
```

```
Number of matches 60818
Number of matches After Lowe's Ratio 4334
Number of Robust matches 3187
```

```
 16%|█▌        | 9/57 [00:40<06:25,  8.02s/it]
```

```
Number of matches 64811
Number of matches After Lowe's Ratio 3400
Number of Robust matches 2156
```

```
Number of matches 60297
Number of matches After Lowe's Ratio 3758
```

```
 18%|█▊        | 10/57 [00:54<07:44,  9.88s/it]
```

```
Number of Robust matches 1856
```

```
 19%|█▉        | 11/57 [01:06<08:05, 10.55s/it]
```

```
Number of matches 46669
Number of matches After Lowe's Ratio 1043
Number of Robust matches 648
```

```
 21%|██        | 12/57 [01:16<07:36, 10.15s/it]
```

```
Number of matches 40665
Number of matches After Lowe's Ratio 3225
Number of Robust matches 2236
```

```
 23%|██▎       | 13/57 [01:24<07:08,  9.74s/it]
```

```
Number of matches 44476
Number of matches After Lowe's Ratio 2367
Number of Robust matches 1757
```

```
 25%|██▍       | 14/57 [01:31<06:24,  8.94s/it]
```

```
Number of matches 19206
Number of matches After Lowe's Ratio 663
Number of matches After Lowe's Ratio 663
Number of Robust matches 310


 26%|██▋       | 15/57 [01:35<05:05,  7.28s/it]

Number of matches 27962
Number of matches After Lowe's Ratio 871
Number of Robust matches 354


 28%|██▊       | 16/57 [01:41<04:40,  6.85s/it]

Number of matches 42909
Number of matches After Lowe's Ratio 970
Number of Robust matches 636



Number of matches 54384
Number of matches After Lowe's Ratio 1523
 30%|██▉       | 17/57 [01:51<05:10,  7.76s/it]

Number of Robust matches 1318


 32%|███▏      | 18/57 [02:03<05:57,  9.15s/it]

Number of matches 56133
Number of matches After Lowe's Ratio 1407
Number of Robust matches 1066


 33%|███▎      | 19/57 [02:16<06:29, 10.25s/it]

Number of matches 61727
Number of matches After Lowe's Ratio 3398
Number of Robust matches 2811


 35%|███▌      | 20/57 [02:27<06:28, 10.51s/it]

Number of matches 38072
Number of matches After Lowe's Ratio 1876
Number of Robust matches 1233


 37%|███▋      | 21/57 [02:32<05:24,  9.02s/it]

Number of matches 12869
Number of matches After Lowe's Ratio 838
Number of Robust matches 504


 39%|███▊      | 22/57 [02:34<03:57,  6.79s/it]

Number of matches 11410
Number of matches After Lowe's Ratio 916
Number of Robust matches 503


 40%|████      | 23/57 [02:36<02:59,  5.29s/it]

Number of matches 13095
Number of matches After Lowe's Ratio 571
Number of Robust matches 243
```

```
42%|████▏     | 24/57 [02:38<02:19,  4.22s/it]
```

```
Number of matches 14928
Number of matches After Lowe's Ratio 635
Number of Robust matches 180
```

```
44%|████▍     | 25/57 [02:39<01:51,  3.49s/it]
```

```
Number of matches 9038
Number of matches After Lowe's Ratio 483
Number of Robust matches 194
```

```
46%|████▌     | 26/57 [02:40<01:25,  2.77s/it]
```

```
Number of matches 8233
Number of matches After Lowe's Ratio 584
Number of Robust matches 203
```

```
47%|████▋     | 27/57 [02:42<01:08,  2.28s/it]
```

```
Number of matches 13342
Number of matches After Lowe's Ratio 486
Number of Robust matches 103
```

```
49%|████▉     | 28/57 [02:43<01:01,  2.11s/it]
```

```
Number of matches 14930
Number of matches After Lowe's Ratio 687
Number of Robust matches 362
```

```
51%|█████     | 29/57 [02:45<00:57,  2.05s/it]
```

```
Number of matches 15458
Number of matches After Lowe's Ratio 1672
Number of Robust matches 1001
```

```
53%|█████▎    | 30/57 [02:47<00:56,  2.10s/it]
```

```
Number of matches 12511
Number of matches After Lowe's Ratio 707
Number of Robust matches 363
```

```
54%|█████▍    | 31/57 [02:49<00:50,  1.93s/it]
```

```
Number of matches 11215
Number of matches After Lowe's Ratio 777
Number of Robust matches 386
```

```
56%|█████▌    | 32/57 [02:50<00:44,  1.78s/it]
```

```
Number of matches 9819
Number of matches After Lowe's Ratio 373
Number of Robust matches 75
```

```
58%|█████▊    | 33/57 [02:52<00:39,  1.63s/it]
```

```
Number of matches 13353
Number of matches After Lowe's Ratio 658
Number of Robust matches 280
```

```
 60%|██████    | 34/57 [02:53<00:38,  1.66s/it]
```

```
Number of matches 14094
Number of matches After Lowe's Ratio 829
Number of Robust matches 366
```

```
 61%|██████▏   | 35/57 [02:55<00:37,  1.71s/it]
```

```
Number of matches 13989
Number of matches After Lowe's Ratio 961
Number of Robust matches 258
```

```
 63%|██████▎   | 36/57 [02:57<00:36,  1.73s/it]
```

```
Number of matches 14812
Number of matches After Lowe's Ratio 810
Number of Robust matches 344
```

```
 65%|██████▍   | 37/57 [02:59<00:37,  1.89s/it]
```

```
Number of matches 18593
Number of matches After Lowe's Ratio 444
Number of Robust matches 176
```

```
 67%|██████▋   | 38/57 [03:03<00:46,  2.45s/it]
```

```
Number of matches 37236
Number of matches After Lowe's Ratio 1695
Number of Robust matches 828
```

```
 68%|██████▊   | 39/57 [03:11<01:12,  4.05s/it]
```

```
Number of matches 45078
Number of matches After Lowe's Ratio 1976
Number of Robust matches 1147
```

```
 70%|███████   | 40/57 [03:20<01:36,  5.66s/it]
```

```
Number of matches 46007
Number of matches After Lowe's Ratio 1630
Number of Robust matches 1187
```

```
 72%|███████▏  | 41/57 [03:29<01:46,  6.67s/it]
```

```
Number of matches 39321
Number of matches After Lowe's Ratio 2026
Number of Robust matches 1318
```

```
 74%|███████▎  | 42/57 [03:36<01:42,  6.84s/it]
```

```
Number of matches 27661
Number of matches After Lowe's Ratio 1218
Number of Robust matches 811
```

```
 75%|███████▌  | 43/57 [03:41<01:24,  6.05s/it]
```

```
Number of matches 20299
Number of matches After Lowe's Ratio 813
Number of Robust matches 532
```

```
 77%|███████▋  | 44/57 [03:44<01:08,  5.23s/it]
```

```
Number of matches 32210
Number of matches After Lowe's Ratio 1223
Number of Robust matches 514
```

```
 79%|███████▉  | 45/57 [03:50<01:06,  5.51s/it]
```

```
Number of matches 33943
Number of matches After Lowe's Ratio 1828
Number of Robust matches 1163
```

```
 81%|████████  | 46/57 [03:57<01:05,  5.92s/it]
```

```
Number of matches 40203
Number of matches After Lowe's Ratio 888
Number of Robust matches 552
```

```
 82%|████████▏ | 47/57 [04:06<01:08,  6.83s/it]
```

```
Number of matches 49115
Number of matches After Lowe's Ratio 1762
Number of Robust matches 1145
```

```
 84%|████████▍ | 48/57 [04:16<01:09,  7.72s/it]
```

```
Number of matches 41146
Number of matches After Lowe's Ratio 4306
Number of Robust matches 2811
```

```
 86%|████████▌ | 49/57 [04:22<00:59,  7.43s/it]
```

```
Number of matches 22575
Number of matches After Lowe's Ratio 925
Number of Robust matches 464
```

```
 88%|████████▊ | 50/57 [04:26<00:43,  6.26s/it]
```

```
Number of matches 17874
Number of matches After Lowe's Ratio 1679
Number of Robust matches 1054
```

```
 89%|████████▉ | 51/57 [04:29<00:30,  5.15s/it]
```

```
Number of matches 18739
Number of matches After Lowe's Ratio 754
Number of Robust matches 448
```

```
 91%|█████████ | 52/57 [04:31<00:21,  4.39s/it]
```

```
Number of matches 15058
Number of matches After Lowe's Ratio 1411
Number of Robust matches 621
```

```
 93%|█████████▎| 53/57 [04:33<00:14,  3.67s/it]
```

```
Number of matches 16673
Number of matches After Lowe's Ratio 931
Number of Robust matches 564
```

```
 95%|█████████▍| 54/57 [04:36<00:09,  3.29s/it]
```

```
Number of matches 18119
Number of matches After Lowe's Ratio 1101
Number of Robust matches 542
```

```
 96%|█████████▋| 55/57 [04:38<00:06,  3.08s/it]
```

```
Number of matches 17288
Number of matches After Lowe's Ratio 1747
Number of Robust matches 1045
```

```
  0%|          | 0/57 [00:00<?, ?it/s]
```

```
Number of matches 19226
Number of matches After Lowe's Ratio 1099
Number of Robust matches 542
```

```
  2%|▏         | 1/57 [00:01<01:20,  1.43s/it]
```

```
Number of matches 14131
Number of matches After Lowe's Ratio 1065
Number of Robust matches 686
```

```
  4%|▎         | 2/57 [00:03<01:25,  1.56s/it]
```

```
Number of matches 16692
Number of matches After Lowe's Ratio 978
Number of Robust matches 747
```

```
  5%|▌         | 3/57 [00:05<01:36,  1.79s/it]
```

```
Number of matches 17997
Number of matches After Lowe's Ratio 1580
Number of Robust matches 1217
```

```
  7%|▋         | 4/57 [00:07<01:43,  1.96s/it]
```

```
Number of matches 13972
Number of matches After Lowe's Ratio 1519
Number of Robust matches 986
```

```
  9%|▉         | 5/57 [00:09<01:38,  1.89s/it]
```

```
Number of matches 9887
Number of matches After Lowe's Ratio 839
Number of Robust matches 253
```

```
 11%|█         | 6/57 [00:10<01:24,  1.67s/it]
```

```
Number of matches 7825
Number of matches After Lowe's Ratio 605
Number of Robust matches 170
```

```
 12%|█▏        | 7/57 [00:12<01:18,  1.56s/it]
```

```
Number of matches 11551
Number of matches After Lowe's Ratio 590
Number of Robust matches 229
```

```
 14%|█▍        | 8/57 [00:13<01:16,  1.56s/it]
```

```
Number of matches 16175
Number of matches After Lowe's Ratio 671
Number of Robust matches 74
```

```
 16%|█▌        | 9/57 [00:15<01:24,  1.76s/it]
```

```
Number of matches 14972
Number of matches After Lowe's Ratio 468
Number of Robust matches 142
```

```
 18%|█▊        | 10/57 [00:17<01:24,  1.79s/it]
```

```
Number of matches 12516
Number of matches After Lowe's Ratio 597
Number of Robust matches 199
```

```
 19%|█▉        | 11/57 [00:19<01:22,  1.80s/it]
```

```
Number of matches 25810
Number of matches After Lowe's Ratio 681
Number of Robust matches 404
```

```
 21%|██        | 12/57 [00:25<02:19,  3.10s/it]
```

```
Number of matches 55502
Number of matches After Lowe's Ratio 1476
Number of Robust matches 950
```

```
 23%|██▎       | 13/57 [00:38<04:18,  5.88s/it]
```

```
Number of matches 62136
Number of matches After Lowe's Ratio 2830
Number of Robust matches 2159
```

```
 25%|██▍       | 14/57 [00:51<05:43,  7.99s/it]
```

```
Number of matches 67747
Number of matches After Lowe's Ratio 1846
Number of Robust matches 1693
```

```
 26%|██▋       | 15/57 [01:05<06:51,  9.80s/it]
```

```
Number of matches 61223
Number of matches After Lowe's Ratio 76
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1634
Number of Robust matches New 6
```

```
 28%|██▊       | 16/57 [01:18<07:26, 10.88s/it]
```

```
Number of matches 59411
Number of matches After Lowe's Ratio 5685
Number of Robust matches 4418
```

```
 30%|███       | 17/57 [01:30<07:30, 11.25s/it]
```

```
Number of matches 54442
Number of matches After Lowe's Ratio 1720
Number of Robust matches 1376
```

```
 32%|███▏      | 18/57 [01:43<07:36, 11.71s/it]
```

```
Number of matches 64610
Number of matches After Lowe's Ratio 3802
Number of Robust matches 2258
```

```
 33%|███▎      | 19/57 [01:56<07:42, 12.18s/it]
```

```
Number of matches 58528
Number of matches After Lowe's Ratio 2192
Number of Robust matches 1588
```

```
 35%|███▌      | 20/57 [02:08<07:26, 12.08s/it]
```

```
Number of matches 53231
Number of matches After Lowe's Ratio 4676
Number of Robust matches 3277
```

```
 37%|███▋      | 21/57 [02:19<07:07, 11.88s/it]
```

```
Number of matches 50320
Number of matches After Lowe's Ratio 2773
Number of Robust matches 2315
```

```
 39%|███▊      | 22/57 [02:30<06:41, 11.48s/it]
```

```
Number of matches 48272
Number of matches After Lowe's Ratio 4006
Number of Robust matches 3075
```

```
 40%|████      | 23/57 [02:40<06:12, 10.95s/it]
```

```
Number of matches 40745
Number of matches After Lowe's Ratio 3105
Number of Robust matches 2670
```

```
 42%|████▏     | 24/57 [02:47<05:23,  9.82s/it]
 42%|████▏     | 24/57 [02:47<05:23,  9.62s/it]
```

```
Number of matches 29260
Number of matches After Lowe's Ratio 3257
Number of Robust matches 2379
```

```
 44%|████▍     | 25/57 [02:51<04:18,  8.07s/it]
```

```
Number of matches 11583
Number of matches After Lowe's Ratio 348
Number of Robust matches 173
```

```
 46%|████▌     | 26/57 [02:53<03:12,  6.21s/it]
```

```
Number of matches 17151
Number of matches After Lowe's Ratio 553
Number of Robust matches 327
```

```
 47%|████▋     | 27/57 [02:55<02:34,  5.15s/it]
```

```
Number of matches 21293
Number of matches After Lowe's Ratio 1040
Number of Robust matches 515
```

```
 49%|████▉     | 28/57 [02:58<02:11,  4.53s/it]
```

```
Number of matches 16158
Number of matches After Lowe's Ratio 733
Number of Robust matches 274
```

```
 51%|█████     | 29/57 [03:00<01:45,  3.78s/it]
```

```
Number of matches 12234
Number of matches After Lowe's Ratio 506
Number of Robust matches 193
```

```
 53%|█████▎    | 30/57 [03:02<01:23,  3.08s/it]
```

```
Number of matches 8082
Number of matches After Lowe's Ratio 488
Number of Robust matches 95
```

```
 54%|█████▍    | 31/57 [03:03<01:04,  2.48s/it]
```

```
Number of matches 13279
Number of matches After Lowe's Ratio 873
Number of Robust matches 322
```

```
 56%|█████▌    | 32/57 [03:05<00:59,  2.36s/it]
```

```
Number of matches 18886
Number of matches After Lowe's Ratio 1446
Number of Robust matches 924
```

```
 58%|█████▊    | 33/57 [03:08<01:01,  2.58s/it]
```

```
Number of matches 23797
Number of matches After Lowe's Ratio 3322
Number of Robust matches 2460
Number of Robust matches 2469
```

```
 60%|█████▉    | 34/57 [03:12<01:06,  2.88s/it]
```

```
Number of matches 19302
Number of matches After Lowe's Ratio 1854
Number of Robust matches 1057
```

```
 61%|██████▏   | 35/57 [03:15<01:04,  2.91s/it]
```

```
Number of matches 21846
Number of matches After Lowe's Ratio 1062
Number of Robust matches 375
```

```
 63%|██████▎   | 36/57 [03:18<01:04,  3.07s/it]
```

```
Number of matches 19657
Number of matches After Lowe's Ratio 704
Number of Robust matches 282
```

```
 65%|██████▍   | 37/57 [03:22<01:05,  3.26s/it]
```

```
Number of matches 25900
Number of matches After Lowe's Ratio 358
Number of Robust matches 130
```

```
 67%|██████▋   | 38/57 [03:26<01:06,  3.49s/it]
```

```
Number of matches 20217
Number of matches After Lowe's Ratio 317
Number of Robust matches 94
```

```
 68%|██████▊   | 39/57 [03:29<01:00,  3.37s/it]
```

```
Number of matches 21187
Number of matches After Lowe's Ratio 545
Number of Robust matches 322
```

```
 70%|███████   | 40/57 [03:32<00:56,  3.35s/it]
```

```
Number of matches 20515
Number of matches After Lowe's Ratio 1272
Number of Robust matches 946
```

```
 72%|███████▏  | 41/57 [03:36<00:54,  3.42s/it]
```

```
Number of matches 22144
Number of matches After Lowe's Ratio 1351
Number of Robust matches 1061
```

```
 74%|███████▎  | 42/57 [03:39<00:51,  3.47s/it]
```

```
Number of matches 23346
Number of matches After Lowe's Ratio 750
Number of Robust matches 501
```

```
 75%|███████▌  | 43/57 [03:43<00:50,  3.59s/it]
```

```
Number of matches 27246
Number of matches After Lowe's Ratio 758
Number of Robust matches 587
```

```
 77%|███████▋  | 44/57 [03:49<00:55,  4.26s/it]
```

```
Number of matches 45385
Number of matches After Lowe's Ratio 2594
Number of Robust matches 1572
```

```
 79%|███████▉  | 45/57 [04:00<01:13,  6.10s/it]
```

```
Number of matches 54369
Number of matches After Lowe's Ratio 4441
Number of Robust matches 3240
```

```
 81%|████████  | 46/57 [04:11<01:24,  7.68s/it]
```

```
Number of matches 48141
Number of matches After Lowe's Ratio 1705
Number of Robust matches 1562
```

```
 82%|████████▏ | 47/57 [04:22<01:27,  8.76s/it]
```

```
Number of matches 59939
Number of matches After Lowe's Ratio 3176
Number of Robust matches 2204
```

```
 84%|████████▍ | 48/57 [04:34<01:25,  9.52s/it]
```

```
Number of matches 42568
Number of matches After Lowe's Ratio 2544
Number of Robust matches 1877
```

```
 86%|████████▌ | 49/57 [04:42<01:13,  9.22s/it]
```

```
Number of matches 43515
Number of matches After Lowe's Ratio 2313
Number of Robust matches 1151
```

```
 88%|████████▊ | 50/57 [04:50<01:02,  8.91s/it]
```

```
Number of matches 36221
Number of matches After Lowe's Ratio 2600
Number of Robust matches 1392
```

```
 89%|████████▉ | 51/57 [04:57<00:49,  8.18s/it]
```

```
Number of matches 31538
Number of matches After Lowe's Ratio 2955
Number of Robust matches 1720
```

```
 91%|█████████ | 52/57 [05:02<00:36,  7.27s/it]
```

```
Number of matches 23542
Number of matches After Lowe's Ratio 2391
Number of Robust matches 1520
```

```
 93%|█████████▎| 53/57 [05:06<00:25,  6.29s/it]
```

```
Number of matches 22544
Number of matches After Lowe's Ratio 847
Number of Robust matches 548
```

```
 95%|█████████▍| 54/57 [05:10<00:16,  5.52s/it]
```

```
Number of matches 27595
Number of matches After Lowe's Ratio 2310
Number of Robust matches 1378
```

```
 96%|█████████▋| 55/57 [05:14<00:10,  5.21s/it]
```

```
Number of matches 23524
Number of matches After Lowe's Ratio 1497
Number of Robust matches 1122
```

```
 98%|█████████▊| 56/57 [05:18<00:04,  4.93s/it]
```

```
Number of matches 27087
Number of matches After Lowe's Ratio 1065
Number of Robust matches 729
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_sift_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_sift)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_sift_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.006144762039184 57 [s] ... size 0.00608 MB
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_sift_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_sift)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_r
ight_sift_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.008843660354614258 [s] ... size 0.00608 MB
```

In [ ]:

```
del H_left_sift, H_right_sift,keypoints_all_left_sift, keypoints_all_right_sift, descript
ors_all_left_sift, descriptors_all_right_sift, points_all_left_sift, points_all_right_sif
t
```

In [72]:

```
import cv2
```

In [ ]:

```
import pickle
```

```
Fdb = open('all_feat_fast_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_fast = []
descriptors_all_left_fast = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_fast.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_fast.append(keypoints_each)
  descriptors_all_left_fast.append(descrip_each)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-96f292158307> in <module>()
     16     keypoints_each.append(temp_feature)
     17     descrip_each.append(temp_descriptor)
---> 18   points_all_left_fast.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_
each]))
     19   keypoints_all_left_fast.append(keypoints_each)
     20   descriptors_all_left_fast.append(descrip_each)

NameError: name 'points_all_left_fast' is not defined
```

In [ ]:

```
import pickle
Fdb = open('all_feat_fast_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_fast = []
descriptors_all_right_fast = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_fast.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_fast.append(keypoints_each)
  descriptors_all_right_fast.append(descrip_each)
```

In [ ]:

```
H_left_fast = []
H_right_fast = []

num_matches_fast = []
num_good_matches_fast = []

for j in tqdm(range(len(left_files_path))):
  if j==len(left_files_path)-1:
    break
  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_fast[j:j+2][::-1],points_all_left_fast[j:j+2][::-1],descriptors_all_left_fast[j:j+2][::-1],0.9,6)
  H_left_fast.append(H_a)
  num_matches_fast.append(matches)
  num_good_matches_fast.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_fast[j:j+2][::-1],points_all_right_fast[j:j+2][::-1],descriptors_all_right_fast[j:j+2][::-1],0.9,6)
  H_right_fast.append(H_a)
  num_matches_fast.append(matches)
  num_good_matches_fast.append(gd_matches)
```

In [ ]:

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_fast_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_fast)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_fast_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_fast_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_fast)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_fast_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_fast, H_right_fast,keypoints_all_left_fast, keypoints_all_right_fast, descriptors_all_left_fast, descriptors_all_right_fast, points_all_left_fast, points_all_right_fast
```

In [ ]:

In [73]:

```
import pickle
Fdb = open('all_feat_orb_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_orb = []
descriptors_all_left_orb = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_orb.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_orb.append(keypoints_each)
  descriptors_all_left_orb.append(descrip_each)
```

In [74]:

```
import pickle
Fdb = open('all_feat_orb_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_orb = []
descriptors_all_right_orb = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                                _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_orb.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_orb.append(keypoints_each)
  descriptors_all_right_orb.append(descrip_each)
```

In [75]:

```
H_left_orb = []
H_right_orb = []

num_matches_orb = []
num_good_matches_orb = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_o
rb[j:j+2][::-1],points_all_left_orb[j:j+2][::-1],descriptors_all_left_orb[j:j+2][::-1],0
.7)
  H_left_orb.append(H_a)
  num_matches_orb.append(matches)
  num_good_matches_orb.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_orb[j:j+2][::-1],points_all_right_orb[j:j+2][::-1],descriptors_all_right_orb[j:j+2][::-
1],0.7)
  H_right_orb.append(H_a)
  num_matches_orb.append(matches)
  num_good_matches_orb.append(gd_matches)
```

```
  0%|          | 0/57 [00:00<?, ?it/s]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 247
Number of Robust matches 11


Number of matches After Lowe's Ratio New 2280
```

```
  2%|▏         | 1/57 [00:01<01:20,  1.43s/it]
```

```
Number of Robust matches New 8
```

```
  4%|▎         | 2/57 [00:02<01:13,  1.33s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 123
Number of Robust matches 20


Number of matches After Lowe's Ratio New 1855
Number of Robust matches New 14
```

```
  5%|▌         | 3/57 [00:03<01:07,  1.26s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 152
Number of Robust matches 13


Number of matches After Lowe's Ratio New 1949
Number of Robust matches New 11
```

```
  7%|▋         | 4/57 [00:04<01:04,  1.21s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 134
Number of Robust matches 8


Number of matches After Lowe's Ratio New 1871
Number of Robust matches New 19
```

```
  9%|▉         | 5/57 [00:05<00:59,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 179
Number of Robust matches 31
```

```
 11%|█         | 6/57 [00:07<01:02,  1.22s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 81
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1534
Number of Robust matches New 6
```

```
 12%|█▏        | 7/57 [00:08<00:56,  1.14s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 104
Number of Robust matches 31
```

```
 14%|█▍        | 8/57 [00:08<00:52,  1.07s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 129
Number of Robust matches 49
```

```
 16%|█▌        | 9/57 [00:10<00:51,  1.07s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 93
Number of Robust matches 15


Number of matches After Lowe's Ratio New 1741
Number of Robust matches New 6
```

```
 18%|█▊        | 10/57 [00:11<00:50,  1.07s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 100
Number of Robust matches 24


Number of matches After Lowe's Ratio New 1762
Number of Robust matches New 9
```

```
 19%|█▉        | 11/57 [00:12<00:53,  1.17s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 75
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1672
Number of Robust matches New 6
```

```
 21%|██        | 12/57 [00:13<00:49,  1.10s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 110
Number of Robust matches 37
```

```
 23%|██▎       | 13/57 [00:14<00:47,  1.09s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 102
Number of Robust matches 23


Number of matches After Lowe's Ratio New 1674
Number of Robust matches New 7
```

```
 25%|██▍       | 14/57 [00:15<00:47,  1.10s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 80
Number of Robust matches 9


Number of matches After Lowe's Ratio New 1720
Number of Robust matches New 7
```

```
 26%|██▋       | 15/57 [00:16<00:46,  1.10s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 94
Number of Robust matches 6
Number of matches After Lowe's Ratio New 1628
Number of Robust matches New 8

 28%|██▊       | 16/57 [00:18<00:48,  1.18s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 77
Number of Robust matches 17

Number of matches After Lowe's Ratio New 1558
Number of Robust matches New 7

 30%|██▉       | 17/57 [00:19<00:46,  1.16s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 64
Number of Robust matches 5

Number of matches After Lowe's Ratio New 1621
Number of Robust matches New 6

 32%|███▏      | 18/57 [00:20<00:44,  1.14s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 62
Number of Robust matches 4

Number of matches After Lowe's Ratio New 1553
Number of Robust matches New 6

 33%|███▎      | 19/57 [00:21<00:41,  1.09s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 120
Number of Robust matches 38

 35%|███▌      | 20/57 [00:22<00:43,  1.19s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 73
Number of Robust matches 16

Number of matches After Lowe's Ratio New 1589
Number of Robust matches New 6

 37%|███▋      | 21/57 [00:23<00:41,  1.16s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 89
Number of Robust matches 7

Number of matches After Lowe's Ratio New 1636
Number of Robust matches New 7
```

```
 39%|███▉      | 22/57 [00:24<00:40,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 173
Number of Robust matches 16


Number of matches After Lowe's Ratio New 1967
Number of Robust matches New 10
```

```
 40%|████      | 23/57 [00:26<00:38,  1.14s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 141
Number of Robust matches 10


Number of matches After Lowe's Ratio New 1920
Number of Robust matches New 9
```

```
 42%|████▏     | 24/57 [00:27<00:37,  1.14s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 133
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1719
Number of Robust matches New 7
```

```
 44%|████▍     | 25/57 [00:28<00:39,  1.22s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 110
Number of Robust matches 10


Number of matches After Lowe's Ratio New 1724
Number of Robust matches New 9
```

```
 46%|████▌     | 26/57 [00:29<00:36,  1.19s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 237
Number of Robust matches 15


Number of matches After Lowe's Ratio New 2213
Number of Robust matches New 17
```

```
 47%|████▋     | 27/57 [00:30<00:35,  1.17s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 220
Number of Robust matches 12


Number of matches After Lowe's Ratio New 2290
Number of Robust matches New 12
```

```
 49%|████▉     | 28/57 [00:31<00:33,  1.16s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 118
Number of Robust matches 21


Number of matches After Lowe's Ratio New 1722
Number of Robust matches New 12
```

```
 51%|█████     | 29/57 [00:32<00:31,  1.12s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 235
Number of Robust matches 29
```

```
 53%|█████▎    | 30/57 [00:34<00:32,  1.22s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 126
Number of Robust matches 10


Number of matches After Lowe's Ratio New 1826
Number of Robust matches New 7
```

```
 54%|█████▍    | 31/57 [00:35<00:31,  1.20s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 110
Number of Robust matches 12


Number of matches After Lowe's Ratio New 1718
Number of Robust matches New 10
```

```
 56%|█████▌    | 32/57 [00:36<00:29,  1.17s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 228
Number of Robust matches 13


Number of matches After Lowe's Ratio New 2268
Number of Robust matches New 10
```

```
 58%|█████▊    | 33/57 [00:37<00:27,  1.16s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 176
Number of Robust matches 15


Number of matches After Lowe's Ratio New 1972
Number of Robust matches New 11
```

```
 60%|██████    | 34/57 [00:39<00:28,  1.25s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 149
Number of Robust matches 15


Number of matches After Lowe's Ratio New 1839
Number of Robust matches New 16
Number of Robust matches New 10
```

```
 61%|██████    | 35/57 [00:40<00:26,  1.21s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 206
Number of Robust matches 18


Number of matches After Lowe's Ratio New 2106
Number of Robust matches New 9
```

```
 63%|██████▎   | 36/57 [00:41<00:24,  1.19s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 125
Number of Robust matches 13


Number of matches After Lowe's Ratio New 1677
Number of Robust matches New 10
```

```
 65%|██████▍   | 37/57 [00:42<00:23,  1.17s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 92
Number of Robust matches 7


Number of matches After Lowe's Ratio New 1647
Number of Robust matches New 7
```

```
 67%|██████▋   | 38/57 [00:43<00:21,  1.16s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 70
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1595
Number of Robust matches New 6
```

```
 68%|██████▊   | 39/57 [00:45<00:21,  1.20s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 115
Number of Robust matches 46
```

```
 70%|███████   | 40/57 [00:46<00:19,  1.17s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 71
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1522
Number of Robust matches New 6
```

```
 72%|███████▏  | 41/57 [00:47<00:18,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 63
Number of matches After Lowe's Ratio 63
Number of Robust matches 13


Number of matches After Lowe's Ratio New 1425
Number of Robust matches New 6
```

```
 74%|███████▍  | 42/57 [00:48<00:16,  1.10s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 117
Number of Robust matches 42
```

```
 75%|███████▌  | 43/57 [00:49<00:16,  1.18s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 146
Number of Robust matches 27
```

```
 77%|███████▋  | 44/57 [00:50<00:15,  1.16s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 85
Number of Robust matches 9


Number of matches After Lowe's Ratio New 1619
Number of Robust matches New 8
```

```
 79%|███████▉  | 45/57 [00:51<00:13,  1.11s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 153
Number of Robust matches 47
```

```
 81%|████████  | 46/57 [00:52<00:12,  1.12s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 70
Number of Robust matches 7


Number of matches After Lowe's Ratio New 1596
Number of Robust matches New 6
```

```
 82%|████████▏ | 47/57 [00:54<00:11,  1.12s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 90
Number of Robust matches 16


Number of matches After Lowe's Ratio New 1622
Number of Robust matches New 6
```

```
 84%|████████▍ | 48/57 [00:55<00:10,  1.17s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 186
Number of Robust matches 92
```

```
 86%|████████▋ | 49/57 [00:56<00:09,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 89
Number of Robust matches 8


Number of matches After Lowe's Ratio New 1581
Number of Robust matches New 7
```

```
 88%|████████▊ | 50/57 [00:57<00:07,  1.10s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 164
Number of Robust matches 39
```

```
 89%|████████▉ | 51/57 [00:58<00:06,  1.11s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 124
Number of Robust matches 24


Number of matches After Lowe's Ratio New 1600
Number of Robust matches New 7
```

```
 91%|█████████ | 52/57 [00:59<00:05,  1.08s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 179
Number of Robust matches 49
```

```
 93%|█████████▎| 53/57 [01:00<00:04,  1.16s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 175
Number of Robust matches 36
```

```
 95%|█████████▍| 54/57 [01:01<00:03,  1.11s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 193
Number of Robust matches 56
```

```
 96%|█████████▋| 55/57 [01:02<00:02,  1.08s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 131
Number of Robust matches 26
```

```
  0%|          | 0/57 [00:00<?, ?it/s]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 129
Number of Robust matches 28
```

```
  2%|▏         | 1/57 [00:01<00:58,  1.04s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 161
Number of Robust matches 35


  4%|▍         | 2/57 [00:02<01:01,  1.12s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 179
Number of Robust matches 61


  5%|▌         | 3/57 [00:03<00:58,  1.08s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 281
Number of Robust matches 130


  7%|▋         | 4/57 [00:04<00:55,  1.05s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 402
Number of Robust matches 229


  9%|▉         | 5/57 [00:05<00:55,  1.07s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 165
Number of Robust matches 23


Number of matches After Lowe's Ratio New 2053
Number of Robust matches New 10


 11%|█         | 6/57 [00:06<01:00,  1.19s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 225
Number of Robust matches 21


Number of matches After Lowe's Ratio New 2155
Number of Robust matches New 13


 12%|█▏        | 7/57 [00:08<00:58,  1.17s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 121
Number of Robust matches 8


Number of matches After Lowe's Ratio New 1883
Number of Robust matches New 8


 14%|█▍        | 8/57 [00:09<00:57,  1.16s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 142
Number of Robust matches 9


Number of matches After Lowe's Ratio New 1911
Number of Robust matches New 7
Number of Robust matches New 7
```

```
 16%|█▌        | 9/57 [00:10<00:55,  1.16s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 103
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1766
Number of Robust matches New 8
```

```
 18%|█▊        | 10/57 [00:11<00:54,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 109
Number of Robust matches 9


Number of matches After Lowe's Ratio New 1776
Number of Robust matches New 9
```

```
 19%|█▉        | 11/57 [00:12<00:57,  1.24s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 100
Number of Robust matches 15


Number of matches After Lowe's Ratio New 1686
Number of Robust matches New 10
```

```
 21%|██        | 12/57 [00:14<00:54,  1.21s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 64
Number of Robust matches 4


Number of matches After Lowe's Ratio New 1473
Number of Robust matches New 7
```

```
 23%|██▎       | 13/57 [00:15<00:50,  1.14s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 84
Number of Robust matches 25
```

```
 25%|██▍       | 14/57 [00:16<00:48,  1.13s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 74
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1632
Number of Robust matches New 6
```

```
 26%|██▋       | 15/57 [00:17<00:47,  1.13s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 59
Number of matches After Lowe's Ratio 59
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1732
Number of Robust matches New 6
```

```
 28%|██▊       | 16/57 [00:18<00:48,  1.18s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 140
Number of Robust matches 35
```

```
 30%|███       | 17/57 [00:19<00:46,  1.17s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 100
Number of Robust matches 18


Number of matches After Lowe's Ratio New 1642
Number of Robust matches New 6
```

```
 32%|███▏      | 18/57 [00:20<00:44,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 89
Number of Robust matches 15


Number of matches After Lowe's Ratio New 1718
Number of Robust matches New 7
```

```
 33%|███▎      | 19/57 [00:21<00:43,  1.14s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 108
Number of Robust matches 11


Number of matches After Lowe's Ratio New 1756
Number of Robust matches New 8
```

```
 35%|███▌      | 20/57 [00:23<00:43,  1.19s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 148
Number of Robust matches 50
```

```
 37%|███▋      | 21/57 [00:24<00:40,  1.12s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 105
Number of Robust matches 30
```

```
 39%|███▊      | 22/57 [00:25<00:37,  1.08s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 109
Number of Robust matches 39
```

```
40%|████      | 23/57 [00:26<00:35,  1.04s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 92
Number of Robust matches 36
```

```
42%|████▏     | 24/57 [00:27<00:33,  1.03s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 216
Number of Robust matches 108
```

```
44%|████▍     | 25/57 [00:28<00:36,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 115
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1795
Number of Robust matches New 11
```

```
46%|████▌     | 26/57 [00:29<00:35,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 137
Number of Robust matches 13


Number of matches After Lowe's Ratio New 1826
Number of Robust matches New 34
```

```
47%|████▋     | 27/57 [00:30<00:34,  1.16s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 119
Number of Robust matches 12


Number of matches After Lowe's Ratio New 1877
Number of Robust matches New 11
```

```
49%|████▉     | 28/57 [00:32<00:33,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 141
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1955
Number of Robust matches New 8
```

```
51%|█████     | 29/57 [00:33<00:32,  1.15s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 122
Number of Robust matches 22


Number of matches After Lowe's Ratio New 1969
Number of Robust matches New 8
```

```
 53%|█████▎    | 30/57 [00:34<00:33,  1.24s/it]
Number of matches 20000
Number of matches After Lowe's Ratio 180
Number of Robust matches 13


Number of matches After Lowe's Ratio New 2061
Number of Robust matches New 17


 54%|█████▍    | 31/57 [00:35<00:31,  1.21s/it]
Number of matches 20000
Number of matches After Lowe's Ratio 176
Number of Robust matches 19


Number of matches After Lowe's Ratio New 1939
Number of Robust matches New 8


 56%|█████▌    | 32/57 [00:36<00:28,  1.14s/it]
Number of matches 20000
Number of matches After Lowe's Ratio 308
Number of Robust matches 161


 58%|█████▊    | 33/57 [00:37<00:26,  1.09s/it]
Number of matches 20000
Number of matches After Lowe's Ratio 662
Number of Robust matches 461


 60%|█████▉    | 34/57 [00:39<00:26,  1.16s/it]
Number of matches 20000
Number of matches After Lowe's Ratio 268
Number of Robust matches 126


 61%|██████▏   | 35/57 [00:40<00:25,  1.16s/it]
Number of matches 20000
Number of matches After Lowe's Ratio 128
Number of Robust matches 8


Number of matches After Lowe's Ratio New 1812
Number of Robust matches New 8


 63%|██████▎   | 36/57 [00:41<00:24,  1.15s/it]
Number of matches 20000
Number of matches After Lowe's Ratio 90
Number of Robust matches 11


Number of matches After Lowe's Ratio New 1726
Number of Robust matches New 10
```

```
 65%|██████▌   | 37/57 [00:42<00:23,  1.15s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 74
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1598
Number of Robust matches New 7


 67%|██████▋   | 38/57 [00:43<00:21,  1.14s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 101
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1730
Number of Robust matches New 7


 68%|██████▊   | 39/57 [00:45<00:22,  1.24s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 78
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1562
Number of Robust matches New 6


 70%|███████   | 40/57 [00:46<00:20,  1.21s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 107
Number of Robust matches 15


Number of matches After Lowe's Ratio New 1794
Number of Robust matches New 10


 72%|███████▏  | 41/57 [00:47<00:18,  1.15s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 159
Number of Robust matches 57


 74%|███████▎  | 42/57 [00:48<00:17,  1.13s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 88
Number of Robust matches 21


Number of matches After Lowe's Ratio New 1654
Number of Robust matches New 9


 75%|███████▌  | 43/57 [00:49<00:15,  1.13s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 89
Number of Robust matches 18
Number of matches After Lowe's Ratio New 1617
Number of Robust matches New 7


 77%|███████▋  | 44/57 [00:50<00:15,  1.23s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 63
Number of Robust matches 9


Number of matches After Lowe's Ratio New 1538
Number of Robust matches New 6


 79%|███████▉  | 45/57 [00:51<00:13,  1.15s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 180
Number of Robust matches 89


 81%|████████  | 46/57 [00:52<00:12,  1.13s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 91
Number of Robust matches 22


Number of matches After Lowe's Ratio New 1679
Number of Robust matches New 43


 82%|████████▏ | 47/57 [00:53<00:10,  1.09s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 95
Number of Robust matches 26


 84%|████████▍ | 48/57 [00:55<00:10,  1.16s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 98
Number of Robust matches 28


 86%|████████▌ | 49/57 [00:56<00:09,  1.15s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 100
Number of Robust matches 19


Number of matches After Lowe's Ratio New 1573
Number of Robust matches New 6


 88%|████████▊ | 50/57 [00:57<00:07,  1.10s/it]

Number of matches 20000
Number of matches After Lowe's Ratio 124
Number of Robust matches 45


 89%|████████▉ | 51/57 [00:58<00:06,  1.08s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 101
Number of Robust matches 35
```

```
 91%|█████████ | 52/57 [00:59<00:05,  1.06s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 115
Number of Robust matches 43
```

```
 93%|█████████▎| 53/57 [01:00<00:04,  1.19s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 95
Number of Robust matches 8


Number of matches After Lowe's Ratio New 1811
Number of Robust matches New 8
```

```
 95%|█████████▍| 54/57 [01:01<00:03,  1.13s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 265
Number of Robust matches 121
```

```
 96%|█████████▋| 55/57 [01:02<00:02,  1.10s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 118
Number of Robust matches 37
```

```
 98%|█████████▊| 56/57 [01:04<00:01,  1.11s/it]
```

```
Number of matches 20000
Number of matches After Lowe's Ratio 109
Number of Robust matches 7


Number of matches After Lowe's Ratio New 1725
Number of Robust matches New 7
```

In [76]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_orb_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_orb)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_orb_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.005198955535888672 [s] ... size 0.00608 MB
```

In [77]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_orb_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_orb)
f.close()
```

```
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_orb_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.005449056625366211 [s] ... size 0.00608 MB
```

In [78]:

```
del H_left_orb, H_right_orb,keypoints_all_left_orb, keypoints_all_right_orb, descriptors_all_left_orb, descriptors_all_right_orb, points_all_left_orb, points_all_right_orb
```

In [79]:

```
import pickle
Fdb = open('all_feat_kaze_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_kaze = []
descriptors_all_left_kaze = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_kaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_kaze.append(keypoints_each)
  descriptors_all_left_kaze.append(descrip_each)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-79-e1ab6833db38> in <module>()
      1 import pickle
----> 2 Fdb = open('all_feat_kaze_left.dat', 'rb')
      3 kpts_all = pickle.load(Fdb)
      4 Fdb.close()
      5 

FileNotFoundError: [Errno 2] No such file or directory: 'all_feat_kaze_left.dat'
```

In [ ]:

```
import pickle
Fdb = open('all_feat_kaze_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_kaze = []
descriptors_all_right_kaze = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_kaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_kaze.append(keypoints_each)
  descriptors_all_right_kaze.append(descrip_each)
```

In [ ]:

In [ ]:

```
H_left_kaze = []
H_right_kaze = []

num_matches_kaze = []
num_good_matches_kaze = []

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

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

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

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

In [ ]:

```

```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_kaze_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_kaze)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_kaze_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_kaze_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_kaze)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_kaze_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_kaze, H_right_kaze,keypoints_all_left_kaze, keypoints_all_right_kaze, descriptors_all_left_kaze, descriptors_all_right_kaze, points_all_left_kaze, points_all_right_kaze
```

In [80]:

```
import pickle
Fdb = open('all_feat_akaze_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_akaze = []
descriptors_all_left_akaze = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_akaze.append(keypoints_each)
  descriptors_all_left_akaze.append(descrip_each)
```

In [81]:

```
import pickle
Fdb = open('all_feat_akaze_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_akaze = []
descriptors_all_right_akaze = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each])
)
  keypoints_all_right_akaze.append(keypoints_each)
  descriptors_all_right_akaze.append(descrip_each)
```

In [82]:

```
H_left_akaze = []
H_right_akaze = []

num_matches_akaze = []
num_good_matches_akaze = []

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

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

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_akaze[j:j+2][::-1],points_all_right_akaze[j:j+2][::-1],descriptors_all_right_akaze[j:j+
2][::-1])
  H_right_akaze.append(H_a)
  num_matches_akaze.append(matches)
  num_good_matches_akaze.append(gd_matches)
```

```
  2%|▏         | 1/57 [00:02<02:26,  2.62s/it]
```

```
Number of matches 21392
Number of matches After Lowe's Ratio 428
Number of Robust matches 71
```

```
  4%|▎         | 2/57 [00:05<02:25,  2.65s/it]
```

```
Number of matches 34946
Number of matches After Lowe's Ratio 545
Number of Robust matches 173
```

```
  5%|▌         | 3/57 [00:09<02:41,  2.99s/it]
```

```
Number of matches 49891
Number of matches After Lowe's Ratio 1287
Number of Robust matches 290
```

```
  7%|▋         | 4/57 [00:31<07:40,  8.70s/it]
```

```
Number of matches 45880
Number of matches After Lowe's Ratio 557
Number of Robust matches 102
```

```
  9%|▉         | 5/57 [00:50<10:18, 11.89s/it]
```

```
Number of matches 41405
Number of matches After Lowe's Ratio 1003
Number of Robust matches 395
```

```
 11%|█         | 6/57 [01:21<15:00, 17.65s/it]
```

```
Number of matches 73372
Number of matches After Lowe's Ratio 1195
Number of Robust matches 353
```

```
 12%|█▏        | 7/57 [02:08<21:58, 26.37s/it]
```

```
Number of matches 159632
Number of matches After Lowe's Ratio 2327
Number of Robust matches 1197
```

```
 14%|█▍        | 8/57 [02:52<25:51, 31.66s/it]
```

```
Number of matches 203916
Number of matches After Lowe's Ratio 5124
Number of Robust matches 2239
```

```
 16%|█▌        | 9/57 [03:32<27:24, 34.26s/it]
```

```
Number of matches 218845
Number of matches After Lowe's Ratio 2179
Number of Robust matches 834
```

```
 18%|█▊        | 10/57 [03:59<25:13, 32.20s/it]
```

```
Number of matches 207210
Number of matches After Lowe's Ratio 1939
Number of Robust matches 632
```

```
 19%|█▉        | 11/57 [04:20<21:56, 28.62s/it]
```

```
Number of matches 150714
Number of matches After Lowe's Ratio 1246
Number of Robust matches 531
```

```
 21%|██        | 12/57 [04:36<18:36, 24.80s/it]
```

```
Number of matches 132512
Number of matches After Lowe's Ratio 2382
Number of Robust matches 1014
```

```
 23%|██▎       | 13/57 [04:52<16:14, 22.15s/it]
```

```
Number of matches 135136
Number of matches After Lowe's Ratio 917
Number of Robust matches 243
```

```
 25%|██▍       | 14/57 [05:00<12:53, 17.99s/it]
```

```
Number of matches 42452
Number of matches After Lowe's Ratio 385
Number of Robust matches 98
```

```
 26%|██▋       | 15/57 [05:07<10:21, 14.80s/it]
```

```
Number of matches 76232
Number of matches After Lowe's Ratio 713
Number of Robust matches 177
```

```
 28%|██▊       | 16/57 [05:24<10:26, 15.28s/it]
```

```
Number of matches 137000
Number of matches After Lowe's Ratio 948
Number of Robust matches 360
```

```
 30%|██▉       | 17/57 [05:42<10:52, 16.30s/it]
```

```
Number of matches 192868
Number of matches After Lowe's Ratio 1055
Number of Robust matches 434
```

```
 32%|███▏      | 18/57 [06:05<11:55, 18.34s/it]
```

```
Number of matches 200958
Number of matches After Lowe's Ratio 937
Number of Robust matches 258
```

```
 33%|███▎      | 19/57 [06:29<12:39, 19.98s/it]
```

```
Number of matches 198278
Number of matches After Lowe's Ratio 3260
Number of Robust matches 1388
```

```
 35%|███▌      | 20/57 [06:50<12:27, 20.21s/it]
```

```
Number of matches 125123
Number of matches After Lowe's Ratio 755
Number of Robust matches 162
```

```
 37%|███▋      | 21/57 [07:00<10:13, 17.05s/it]
```

```
Number of matches 40443
Number of matches After Lowe's Ratio 821
Number of Robust matches 446
```

```
 39%|███▊      | 22/57 [07:03<07:32, 12.93s/it]
```

```
Number of matches 29084
Number of matches After Lowe's Ratio 599
Number of Robust matches 204
```

```
 40%|████      | 23/57 [07:06<05:34,  9.83s/it]
```

```
Number of matches 36526
Number of matches After Lowe's Ratio 416
Number of Robust matches 130
```

```
 42%|████▏     | 24/57 [07:10<04:34,  8.31s/it]
```

```
Number of matches 49473
Number of matches After Lowe's Ratio 745
Number of Robust matches 247
```

```
 44%|████▍     | 25/57 [07:15<03:46,  7.07s/it]
```

```
Number of matches 32594
Number of matches After Lowe's Ratio 623
Number of Robust matches 232
```

```
 46%|████▌     | 26/57 [07:18<03:05,  5.98s/it]
```

```
Number of matches 28939
Number of matches After Lowe's Ratio 491
Number of Robust matches 59
```

```
 47%|████▋     | 27/57 [07:21<02:30,  5.02s/it]
```

```
Number of matches 37709
Number of matches After Lowe's Ratio 348
Number of Robust matches 22


Number of matches After Lowe's Ratio New 2300
Number of Robust matches New 11
```

```
 49%|████▉     | 28/57 [07:24<02:13,  4.61s/it]
```

```
Number of matches 41480
Number of matches After Lowe's Ratio 866
Number of Robust matches 369
```

```
 51%|█████     | 29/57 [07:30<02:15,  4.83s/it]
```

```
Number of matches 42017
Number of matches After Lowe's Ratio 1287
Number of Robust matches 358


 53%|█████▎    | 30/57 [07:34<02:06,  4.67s/it]

Number of matches 39692
Number of matches After Lowe's Ratio 724
Number of Robust matches 177


 54%|█████▍    | 31/57 [07:39<02:01,  4.66s/it]

Number of matches 35072
Number of matches After Lowe's Ratio 761
Number of Robust matches 259


 56%|█████▌    | 32/57 [07:41<01:42,  4.08s/it]

Number of matches 28499
Number of matches After Lowe's Ratio 440
Number of Robust matches 33


 58%|█████▊    | 33/57 [07:44<01:25,  3.55s/it]

Number of matches 35396
Number of matches After Lowe's Ratio 502
Number of Robust matches 176


 60%|█████▉    | 34/57 [07:48<01:26,  3.76s/it]

Number of matches 39678
Number of matches After Lowe's Ratio 982
Number of Robust matches 432


 61%|██████▏   | 35/57 [07:52<01:22,  3.74s/it]

Number of matches 36245
Number of matches After Lowe's Ratio 825
Number of Robust matches 157


 63%|██████▎   | 36/57 [07:56<01:24,  4.01s/it]

Number of matches 45085
Number of matches After Lowe's Ratio 735
Number of Robust matches 248


 65%|██████▍   | 37/57 [08:02<01:28,  4.40s/it]

Number of matches 50480
Number of matches After Lowe's Ratio 342
Number of Robust matches 83


 67%|██████▋   | 38/57 [08:12<01:58,  6.23s/it]

Number of matches 105934
Number of matches After Lowe's Ratio 1751
Number of Robust matches 588
```

```
 68%|██████▊   | 39/57 [08:26<02:33,  8.54s/it]
```

```
Number of matches 148296
Number of matches After Lowe's Ratio 1345
Number of Robust matches 619
```

```
 70%|███████   | 40/57 [08:41<02:59, 10.54s/it]
```

```
Number of matches 150059
Number of matches After Lowe's Ratio 1272
Number of Robust matches 581
```

```
 72%|███████▏  | 41/57 [08:55<03:06, 11.63s/it]
```

```
Number of matches 148158
Number of matches After Lowe's Ratio 1349
Number of Robust matches 571
```

```
 74%|███████▎  | 42/57 [09:07<02:54, 11.61s/it]
```

```
Number of matches 75872
Number of matches After Lowe's Ratio 453
Number of Robust matches 127
```

```
 75%|███████▌  | 43/57 [09:15<02:26, 10.47s/it]
```

```
Number of matches 55014
Number of matches After Lowe's Ratio 524
Number of Robust matches 192
```

```
 77%|███████▋  | 44/57 [09:25<02:13, 10.24s/it]
```

```
Number of matches 83496
Number of matches After Lowe's Ratio 1146
Number of Robust matches 558
```

```
 79%|███████▉  | 45/57 [09:36<02:07, 10.63s/it]
```

```
Number of matches 108739
Number of matches After Lowe's Ratio 2057
Number of Robust matches 1102
```

```
 81%|████████  | 46/57 [09:51<02:11, 11.94s/it]
```

```
Number of matches 150038
Number of matches After Lowe's Ratio 748
Number of Robust matches 291
```

```
 82%|████████▏ | 47/57 [10:08<02:15, 13.53s/it]
```

```
Number of matches 157375
Number of matches After Lowe's Ratio 1858
Number of Robust matches 716
```

```
 84%|████████▍ | 48/57 [10:23<02:04, 13.88s/it]
 84%|████████▍ | 48/57 [10:23<02:01, 13.66s/it]
```

```
Number of matches 122095
Number of matches After Lowe's Ratio 3055
Number of Robust matches 1559
```

```
 86%|████████▌ | 49/57 [10:33<01:41, 12.71s/it]
```

```
Number of matches 63855
Number of matches After Lowe's Ratio 630
Number of Robust matches 222
```

```
 88%|████████▊ | 50/57 [10:42<01:21, 11.61s/it]
```

```
Number of matches 63866
Number of matches After Lowe's Ratio 1781
Number of Robust matches 891
```

```
 89%|████████▉ | 51/57 [10:53<01:07, 11.31s/it]
```

```
Number of matches 80814
Number of matches After Lowe's Ratio 939
Number of Robust matches 346
```

```
 91%|█████████ | 52/57 [10:59<00:49,  9.97s/it]
```

```
Number of matches 47961
Number of matches After Lowe's Ratio 1490
Number of Robust matches 646
```

```
 93%|█████████▎| 53/57 [11:06<00:35,  8.82s/it]
```

```
Number of matches 41086
Number of matches After Lowe's Ratio 785
Number of Robust matches 330
```

```
 95%|█████████▍| 54/57 [11:11<00:23,  7.72s/it]
```

```
Number of matches 54203
Number of matches After Lowe's Ratio 1214
Number of Robust matches 452
```

```
 96%|█████████▋| 55/57 [11:19<00:15,  7.75s/it]
```

```
Number of matches 55640
Number of matches After Lowe's Ratio 2092
Number of Robust matches 1075
```

```
  0%|          | 0/57 [00:00<?, ?it/s]
```

```
Number of matches 67680
Number of matches After Lowe's Ratio 902
Number of Robust matches 302
```

```
  2%|▏         | 1/57 [00:04<03:50,  4.11s/it]
```

```
Number of matches 45250
Number of matches After Lowe's Ratio 1560
Number of Robust matches 980
Number of Robust matches 980
```

```
  4%|▎         | 2/57 [00:11<04:37,  5.04s/it]
```

```
Number of matches 60084
Number of matches After Lowe's Ratio 1404
Number of Robust matches 705
```

```
  5%|▌         | 3/57 [00:17<04:55,  5.48s/it]
```

```
Number of matches 46835
Number of matches After Lowe's Ratio 2717
Number of Robust matches 1582
```

```
  7%|▋         | 4/57 [00:23<04:57,  5.61s/it]
```

```
Number of matches 37394
Number of matches After Lowe's Ratio 2734
Number of Robust matches 1648
```

```
  9%|▉         | 5/57 [00:26<04:08,  4.78s/it]
```

```
Number of matches 21633
Number of matches After Lowe's Ratio 529
Number of Robust matches 182
```

```
 11%|█         | 6/57 [00:28<03:12,  3.78s/it]
```

```
Number of matches 20583
Number of matches After Lowe's Ratio 598
Number of Robust matches 175
```

```
 12%|█▏        | 7/57 [00:29<02:38,  3.18s/it]
```

```
Number of matches 34847
Number of matches After Lowe's Ratio 507
Number of Robust matches 126
```

```
 14%|█▍        | 8/57 [00:34<03:03,  3.74s/it]
```

```
Number of matches 49761
Number of matches After Lowe's Ratio 628
Number of Robust matches 109
```

```
 16%|█▌        | 9/57 [00:40<03:26,  4.30s/it]
```

```
Number of matches 44528
Number of matches After Lowe's Ratio 371
Number of Robust matches 105
```

```
 18%|█▊        | 10/57 [00:46<03:50,  4.90s/it]
```

```
Number of matches 48262
Number of matches After Lowe's Ratio 1025
Number of Robust matches 651
```

```
 19%|█▉        | 11/57 [00:54<04:29,  5.85s/it]
```

```
Number of matches 70408
Number of matches After Lowe's Ratio 631
Number of Robust matches 234
```

```
 21%|██        | 12/57 [01:13<07:09,  9.55s/it]
```

```
Number of matches 170206
Number of matches After Lowe's Ratio 1236
Number of Robust matches 557
```

```
 23%|██▎       | 13/57 [01:32<09:14, 12.60s/it]
```

```
Number of matches 172776
Number of matches After Lowe's Ratio 1620
Number of Robust matches 895
```

```
 25%|██▍       | 14/57 [01:58<11:53, 16.58s/it]
```

```
Number of matches 213394
Number of matches After Lowe's Ratio 1337
Number of Robust matches 590



Number of matches 147543
Number of matches After Lowe's Ratio 503
Number of Robust matches 5


Number of matches After Lowe's Ratio New 6439
```

```
 26%|██▋       | 15/57 [02:18<12:21, 17.65s/it]
```

```
Number of Robust matches New 6
```

```
 28%|██▊       | 16/57 [02:40<12:57, 18.97s/it]
```

```
Number of matches 204895
Number of matches After Lowe's Ratio 3417
Number of Robust matches 1225
```

```
 30%|██▉       | 17/57 [03:03<13:18, 19.96s/it]
```

```
Number of matches 178311
Number of matches After Lowe's Ratio 1275
Number of Robust matches 510
```

```
 32%|███▏      | 18/57 [03:26<13:43, 21.12s/it]
```

```
Number of matches 222798
Number of matches After Lowe's Ratio 2881
Number of Robust matches 801
```

```
 33%|███▎      | 19/57 [03:52<14:08, 22.34s/it]
```

```
Number of matches 209287
Number of matches After Lowe's Ratio 1628
Number of Robust matches 536
```

```
 35%|███▌      | 20/57 [04:14<13:50, 22.44s/it]
```

```
Number of matches 182819
Number of matches After Lowe's Ratio 2562
Number of Robust matches 874
```

```
 37%|███▋      | 21/57 [04:35<13:11, 21.98s/it]
```

```
Number of matches 170898
Number of matches After Lowe's Ratio 2686
Number of Robust matches 1253
```

```
 39%|███▊      | 22/57 [04:54<12:15, 21.02s/it]
```

```
Number of matches 161422
Number of matches After Lowe's Ratio 2117
Number of Robust matches 772
```

```
 40%|████      | 23/57 [05:09<10:58, 19.38s/it]
```

```
Number of matches 119059
Number of matches After Lowe's Ratio 1032
Number of Robust matches 439
```

```
Number of matches 86521
Number of matches After Lowe's Ratio 4878
```

```
 42%|████▏     | 24/57 [05:21<09:17, 16.91s/it]
```

```
Number of Robust matches 2526
```

```
 44%|████▍     | 25/57 [05:26<07:08, 13.38s/it]
```

```
Number of matches 22652
Number of matches After Lowe's Ratio 164
Number of Robust matches 41
```

```
 46%|████▌     | 26/57 [05:28<05:12, 10.08s/it]
```

```
Number of matches 46301
Number of matches After Lowe's Ratio 428
Number of Robust matches 79
```

```
 47%|████▋     | 27/57 [05:35<04:31,  9.05s/it]
```

```
Number of matches 52730
Number of matches After Lowe's Ratio 728
Number of Robust matches 230
```

```
 49%|████▉     | 28/57 [05:41<03:55,  8.12s/it]
```

```
Number of matches 43767
Number of matches After Lowe's Ratio 498
Number of Robust matches 117
```

```
 51%|█████     | 29/57 [05:45<03:16,  7.01s/it]
```

```
Number of matches 27831
Number of matches After Lowe's Ratio 309
Number of Robust matches 92
```

```
 53%|█████▎    | 30/57 [05:47<02:26,  5.44s/it]
```

```
Number of matches 18712
Number of matches After Lowe's Ratio 428
Number of Robust matches 99
```

```
 54%|█████▍    | 31/57 [05:49<01:51,  4.30s/it]
```

```
Number of matches 34386
Number of matches After Lowe's Ratio 732
Number of Robust matches 225
```

```
 56%|█████▌    | 32/57 [05:54<01:54,  4.56s/it]
```

```
Number of matches 60772
Number of matches After Lowe's Ratio 2223
Number of Robust matches 1215
```

```
 58%|█████▊    | 33/57 [06:03<02:26,  6.12s/it]
```

```
Number of matches 78017
Number of matches After Lowe's Ratio 7177
Number of Robust matches 5219
```

```
 60%|█████▉    | 34/57 [06:11<02:30,  6.54s/it]
```

```
Number of matches 58845
Number of matches After Lowe's Ratio 2913
Number of Robust matches 1555
```

```
 61%|██████▏   | 35/57 [06:20<02:42,  7.38s/it]
```

```
Number of matches 74646
Number of matches After Lowe's Ratio 1419
Number of Robust matches 527
```

```
 63%|██████▎   | 36/57 [06:30<02:47,  7.98s/it]
```

```
Number of matches 71294
Number of matches After Lowe's Ratio 382
Number of Robust matches 67
```

```
 65%|██████▍   | 37/57 [06:40<02:51,  8.58s/it]
```

```
Number of matches 85897
Number of matches After Lowe's Ratio 335
Number of Robust matches 73
```

```
 67%|██████▋   | 38/57 [06:48<02:44,  8.63s/it]
```

```
Number of matches 61601
Number of matches After Lowe's Ratio 425
Number of Robust matches 157
```

```
 68%|██████▊   | 39/57 [06:56<02:30,  8.34s/it]
```

```
Number of matches 58739
Number of matches After Lowe's Ratio 557
Number of Robust matches 291
```

```
 70%|███████   | 40/57 [07:02<02:10,  7.66s/it]
```

```
Number of matches 31170
Number of matches After Lowe's Ratio 755
Number of Robust matches 490
```

```
 72%|███████▏  | 41/57 [07:06<01:44,  6.51s/it]
```

```
Number of matches 62968
Number of matches After Lowe's Ratio 1522
Number of Robust matches 896
```

```
 74%|███████▎  | 42/57 [07:16<01:53,  7.55s/it]
```

```
Number of matches 73009
Number of matches After Lowe's Ratio 857
Number of Robust matches 553
```

```
 75%|███████▌  | 43/57 [07:26<01:53,  8.14s/it]
```

```
Number of matches 76100
Number of matches After Lowe's Ratio 584
Number of Robust matches 233
```

```
 77%|███████▋  | 44/57 [07:40<02:09,  9.95s/it]
```

```
Number of matches 130183
Number of matches After Lowe's Ratio 1802
Number of Robust matches 765
```

```
 79%|███████▉  | 45/57 [07:58<02:29, 12.43s/it]
```

```
Number of matches 163158
Number of matches After Lowe's Ratio 4230
Number of Robust matches 2178
```

```
 81%|████████  | 46/57 [08:17<02:39, 14.55s/it]
```

```
Number of matches 163180
Number of matches After Lowe's Ratio 1128
Number of Robust matches 441
```

```
 82%|████████▏ | 47/57 [08:40<02:48, 16.88s/it]
```

```
Number of matches 212402
Number of matches After Lowe's Ratio 2410
Number of Robust matches 921
```

```
 84%|████████▍ | 48/57 [09:00<02:41, 17.96s/it]
```

```
Number of matches 146601
Number of matches After Lowe's Ratio 2570
Number of Robust matches 1015
```

```
 86%|████████▋  | 49/57 [09:15<02:16, 17.10s/it]
```

```
Number of matches 141297
Number of matches After Lowe's Ratio 2081
Number of Robust matches 854
```

```
 88%|████████▊  | 50/57 [09:30<01:55, 16.52s/it]
```

```
Number of matches 127220
Number of matches After Lowe's Ratio 1751
Number of Robust matches 693
```

```
 89%|████████▉  | 51/57 [09:44<01:33, 15.60s/it]
```

```
Number of matches 114736
Number of matches After Lowe's Ratio 2464
Number of Robust matches 1043
```

```
 91%|█████████  | 52/57 [09:54<01:09, 13.96s/it]
```

```
Number of matches 68330
Number of matches After Lowe's Ratio 3550
Number of Robust matches 1621
```

```
 93%|█████████▎ | 53/57 [10:02<00:49, 12.30s/it]
```

```
Number of matches 73768
Number of matches After Lowe's Ratio 1133
Number of Robust matches 510
```

```
 95%|█████████▍ | 54/57 [10:12<00:34, 11.61s/it]
```

```
Number of matches 81749
Number of matches After Lowe's Ratio 2437
Number of Robust matches 1369
```

```
 96%|█████████▋| 55/57 [10:20<00:21, 10.51s/it]
```

```
Number of matches 49336
Number of matches After Lowe's Ratio 991
Number of Robust matches 610
```

```
 98%|█████████▊| 56/57 [10:30<00:10, 10.30s/it]
```

```
Number of matches 87173
Number of matches After Lowe's Ratio 912
Number of Robust matches 528
```

In [83]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_akaze_40.h5','w')
```

```
t0=time.time()
f.create_dataset('data',data=H_left_akaze)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_akaze_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.005347490310668945 [s] ... size 0.00608 MB
```

In [84]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_akaze_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_akaze)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_r
ight_akaze_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.004068136215209961 [s] ... size 0.00608 MB
```

In [85]:

```
del H_left_akaze, H_right_akaze,keypoints_all_left_akaze, keypoints_all_right_akaze, desc
riptors_all_left_akaze, descriptors_all_right_akaze, points_all_left_akaze, points_all_ri
ght_akaze
```

In [86]:

```
import pickle
Fdb = open('all_feat_star_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_star = []
descriptors_all_left_brief = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_star.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_star.append(keypoints_each)
  descriptors_all_left_brief.append(descrip_each)
```

In [87]:

```
import pickle
Fdb = open('all_feat_star_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_star = []
descriptors_all_right_brief = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_star.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_star.append(keypoints_each)
  descriptors_all_right_brief.append(descrip_each)
```

In [88]:

```
H_left_brief = []
H_right_brief = []

num_matches_briefstar = []
num_good_matches_briefstar = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_star[j:j+2][::-1],points_all_left_star[j:j+2][::-1],descriptors_all_left_brief[j:j+2][::-1])
  H_left_brief.append(H_a)
  num_matches_briefstar.append(matches)
  num_good_matches_briefstar.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_star[j:j+2][::-1],points_all_right_star[j:j+2][::-1],descriptors_all_right_brief[j:j+2][::-1])
  H_right_brief.append(H_a)
  num_matches_briefstar.append(matches)
  num_good_matches_briefstar.append(gd_matches)
```

```
  2%|▏         | 1/57 [00:00<00:24,  2.26it/s]
```

```
Number of matches 7297
Number of matches After Lowe's Ratio 185
Number of Robust matches 10


Number of matches After Lowe's Ratio New 858
Number of Robust matches New 12
```

```
  4%|▎         | 2/57 [00:00<00:23,  2.34it/s]
```

```
Number of matches 10143
Number of matches After Lowe's Ratio 259
Number of Robust matches 7


Number of matches After Lowe's Ratio New 1183
Number of Robust matches New 9
```

```
  5%|▌         | 3/57 [00:01<00:32,  1.65it/s]
```

```
Number of matches 15230
Number of matches After Lowe's Ratio 389
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1719
Number of Robust matches New 12
```

```
  7%|▋         | 4/57 [00:02<00:32,  1.64it/s]
```

```
Number of matches 14423
Number of matches 14423
Number of matches After Lowe's Ratio 428
Number of Robust matches 7


Number of matches After Lowe's Ratio New 1731
Number of Robust matches New 17


  9%|█         | 5/57 [00:02<00:30,  1.73it/s]

Number of matches 13455
Number of matches After Lowe's Ratio 610
Number of Robust matches 91


 11%|█         | 6/57 [00:03<00:30,  1.65it/s]

Number of matches 32334
Number of matches After Lowe's Ratio 924
Number of Robust matches 76



Number of matches 87323
Number of matches After Lowe's Ratio 1506
Number of Robust matches 6


Number of matches After Lowe's Ratio New 8931

 12%|█▏        | 7/57 [00:07<01:20,  1.60s/it]

Number of Robust matches New 7


 14%|█▍        | 8/57 [00:16<03:12,  3.92s/it]

Number of matches 113582
Number of matches After Lowe's Ratio 3991
Number of Robust matches 1040


 16%|█▌        | 9/57 [00:27<04:37,  5.78s/it]

Number of matches 122179
Number of matches After Lowe's Ratio 2967
Number of Robust matches 228


 18%|█▊        | 10/57 [00:36<05:25,  6.93s/it]

Number of matches 109821
Number of matches After Lowe's Ratio 2793
Number of Robust matches 228


 19%|█▉        | 11/57 [00:43<05:19,  6.95s/it]

Number of matches 76251
Number of matches After Lowe's Ratio 1785
Number of Robust matches 110



Number of matches 68024
Number of matches After Lowe's Ratio 1182
Number of Robust matches 5
Number of matches After Lowe's Ratio New 6441

 21%|██        | 12/57 [00:49<05:02,  6.71s/it]

Number of Robust matches New 6



Number of matches 70468
Number of matches After Lowe's Ratio 1062
Number of Robust matches 8


Number of matches After Lowe's Ratio New 6303

 23%|██▎       | 13/57 [00:56<04:49,  6.57s/it]

Number of Robust matches New 6



 25%|██▍       | 14/57 [00:58<03:51,  5.39s/it]

Number of matches 14429
Number of matches After Lowe's Ratio 250
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1414
Number of Robust matches New 6



Number of matches 36408
Number of matches After Lowe's Ratio 636
Number of Robust matches 5


Number of matches After Lowe's Ratio New 3499

 26%|██▋       | 15/57 [00:59<02:49,  4.03s/it]

Number of Robust matches New 6



Number of matches 67123
Number of matches After Lowe's Ratio 1083
Number of Robust matches 5


Number of matches After Lowe's Ratio New 6411

 28%|██▊       | 16/57 [01:03<02:43,  3.98s/it]

Number of Robust matches New 6



Number of matches 100522
Number of matches After Lowe's Ratio 1342
Number of Robust matches 5


Number of matches After Lowe's Ratio New 8244

 30%|██▉       | 17/57 [01:11<03:28,  5.21s/it]

Number of Robust matches New 6



Number of matches 108912
Number of matches After Lowe's Ratio 1446
Number of matches After Lowe's Ratio 1110
Number of Robust matches 6


Number of matches After Lowe's Ratio New 9265
 32%|███▏      | 18/57 [01:20<04:11,  6.45s/it]
Number of Robust matches New 6



Number of matches 114803
Number of matches After Lowe's Ratio 1510
Number of Robust matches 6


Number of matches After Lowe's Ratio New 9705
 33%|███▎      | 19/57 [01:29<04:33,  7.19s/it]
Number of Robust matches New 7



Number of matches 68405
Number of matches After Lowe's Ratio 990
Number of Robust matches 5


Number of matches After Lowe's Ratio New 6017
 35%|███▌      | 20/57 [01:36<04:21,  7.07s/it]
Number of Robust matches New 6


 37%|███▋      | 21/57 [01:39<03:25,  5.71s/it]
Number of matches 12444
Number of matches After Lowe's Ratio 233
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1257
Number of Robust matches New 9


 39%|███▊      | 22/57 [01:39<02:24,  4.14s/it]
Number of matches 8712
Number of matches After Lowe's Ratio 217
Number of Robust matches 8


Number of matches After Lowe's Ratio New 986
Number of Robust matches New 14


 40%|████      | 23/57 [01:39<01:42,  3.02s/it]
Number of matches 10663
Number of matches After Lowe's Ratio 321
Number of Robust matches 8


Number of matches After Lowe's Ratio New 1294
Number of Robust matches New 9


 42%|████▏     | 24/57 [01:40<01:14,  2.26s/it]
```

```
Number of matches 14292
Number of matches After Lowe's Ratio 580
Number of Robust matches 20


Number of matches After Lowe's Ratio New 1942
Number of Robust matches New 13
```

```
 44%|████▍     | 25/57 [01:40<00:55,  1.72s/it]
```

```
Number of matches 8966
Number of matches After Lowe's Ratio 361
Number of Robust matches 59
```

```
 46%|████▌     | 26/57 [01:41<00:40,  1.32s/it]
```

```
Number of matches 7671
Number of matches After Lowe's Ratio 155
Number of Robust matches 7


Number of matches After Lowe's Ratio New 773
Number of Robust matches New 12
```

```
 47%|████▋     | 27/57 [01:42<00:35,  1.18s/it]
```

```
Number of matches 13285
Number of matches After Lowe's Ratio 272
Number of Robust matches 45
```

```
 49%|████▉     | 28/57 [01:42<00:28,  1.01it/s]
```

```
Number of matches 14584
Number of matches After Lowe's Ratio 309
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1531
Number of Robust matches New 7
```

```
 51%|█████     | 29/57 [01:43<00:24,  1.14it/s]
```

```
Number of matches 15375
Number of matches After Lowe's Ratio 324
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1613
Number of Robust matches New 6
```

```
 53%|█████▎    | 30/57 [01:43<00:20,  1.31it/s]
```

```
Number of matches 13188
Number of matches After Lowe's Ratio 516
Number of Robust matches 25
```

```
 54%|█████▍    | 31/57 [01:44<00:17,  1.49it/s]
```

```
Number of matches 12670
Number of matches After Lowe's Ratio 317
Number of Robust matches 34
```

```
 56%|█████▌    | 32/57 [01:44<00:15,  1.64it/s]
```

```
Number of matches 9910
Number of matches After Lowe's Ratio 188
Number of Robust matches 10


Number of matches After Lowe's Ratio New 987
Number of Robust matches New 20
```

```
 58%|█████▊    | 33/57 [01:45<00:13,  1.80it/s]
```

```
Number of matches 10849
Number of matches After Lowe's Ratio 281
Number of Robust matches 10


Number of matches After Lowe's Ratio New 1187
Number of Robust matches New 12
```

```
 60%|█████▉    | 34/57 [01:46<00:15,  1.53it/s]
```

```
Number of matches 11248
Number of matches After Lowe's Ratio 436
Number of Robust matches 82
```

```
 61%|██████▏   | 35/57 [01:46<00:13,  1.67it/s]
```

```
Number of matches 11853
Number of matches After Lowe's Ratio 265
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1337
Number of Robust matches New 7
```

```
 63%|██████▎   | 36/57 [01:47<00:12,  1.72it/s]
```

```
Number of matches 15772
Number of matches After Lowe's Ratio 312
Number of Robust matches 12


Number of matches After Lowe's Ratio New 1521
Number of Robust matches New 26
```

```
 65%|██████▍   | 37/57 [01:47<00:12,  1.65it/s]
```

```
Number of matches 18403
Number of matches After Lowe's Ratio 385
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1955
Number of Robust matches New 7



Number of matches 57922
Number of matches After Lowe's Ratio 1142
Number of Robust matches 6
Number of Robust matches 6


Number of matches After Lowe's Ratio New 6039
 67%|██████▋   | 38/57 [01:49<00:21,  1.11s/it]
Number of Robust matches New 7



 68%|██████▊   | 39/57 [01:56<00:48,  2.70s/it]
Number of matches 81181
Number of matches After Lowe's Ratio 1162
Number of Robust matches 122




Number of matches 83694
Number of matches After Lowe's Ratio 955
Number of Robust matches 5


Number of matches After Lowe's Ratio New 6554
 70%|███████   | 40/57 [02:03<01:08,  4.05s/it]
Number of Robust matches New 6




Number of matches 75269
Number of matches After Lowe's Ratio 920
Number of Robust matches 5


Number of matches After Lowe's Ratio New 6079
 72%|███████▏  | 41/57 [02:10<01:17,  4.81s/it]
Number of Robust matches New 7



 74%|███████▎  | 42/57 [02:13<01:05,  4.39s/it]
Number of matches 31615
Number of matches After Lowe's Ratio 628
Number of Robust matches 34



 75%|███████▌  | 43/57 [02:15<00:49,  3.52s/it]
Number of matches 22444
Number of matches After Lowe's Ratio 453
Number of Robust matches 8


Number of matches After Lowe's Ratio New 2462
Number of Robust matches New 7




Number of matches 34223
Number of matches After Lowe's Ratio 670
Number of Robust matches 7


Number of matches After Lowe's Ratio New 3707
 77%|███████▋  | 44/57 [02:17<00:39,  3.04s/it]
Number of Robust matches New 8
```

```
 79%|███████▉  | 45/57 [02:19<00:34,  2.88s/it]

Number of matches 49341
Number of matches After Lowe's Ratio 1412
Number of Robust matches 269


 81%|████████  | 46/57 [02:24<00:40,  3.66s/it]

Number of matches 74154
Number of matches After Lowe's Ratio 1109
Number of Robust matches 77



Number of matches 83587
Number of matches After Lowe's Ratio 1181
Number of Robust matches 5


Number of matches After Lowe's Ratio New 7152
 82%|████████▏ | 47/57 [02:31<00:45,  4.56s/it]
Number of Robust matches New 6



Number of matches 63636
Number of matches After Lowe's Ratio 945
Number of Robust matches 5


Number of matches After Lowe's Ratio New 5569
 84%|████████▍ | 48/57 [02:37<00:45,  5.01s/it]
Number of Robust matches New 5


 86%|████████▌ | 49/57 [02:40<00:35,  4.42s/it]

Number of matches 27082
Number of matches After Lowe's Ratio 562
Number of Robust matches 29


 88%|████████▊ | 50/57 [02:42<00:24,  3.52s/it]

Number of matches 25313
Number of matches After Lowe's Ratio 629
Number of Robust matches 11


Number of matches After Lowe's Ratio New 2982
Number of Robust matches New 15



Number of matches 37309
Number of matches After Lowe's Ratio 665
Number of Robust matches 14


Number of matches After Lowe's Ratio New 3583
 89%|████████▉ | 51/57 [02:44<00:18,  3.15s/it]
Number of Robust matches New 13
```

```
 91%|█████████ | 52/57 [02:46<00:13,  2.72s/it]
```

```
Number of matches 19172
Number of matches After Lowe's Ratio 288
Number of Robust matches 9


Number of matches After Lowe's Ratio New 1756
Number of Robust matches New 7
```

```
 93%|█████████▎| 53/57 [02:47<00:08,  2.16s/it]
```

```
Number of matches 15509
Number of matches After Lowe's Ratio 312
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1683
Number of Robust matches New 7
```

```
 95%|█████████▍| 54/57 [02:47<00:05,  1.69s/it]
```

```
Number of matches 23101
Number of matches After Lowe's Ratio 540
Number of Robust matches 32
```

```
 96%|█████████▋| 55/57 [02:49<00:03,  1.67s/it]
```

```
Number of matches 23117
Number of matches After Lowe's Ratio 398
Number of Robust matches 9


Number of matches After Lowe's Ratio New 2259
Number of Robust matches New 18
```

```
  0%|          | 0/57 [00:00<?, ?it/s]
```

```
Number of matches 27749
Number of matches After Lowe's Ratio 697
Number of Robust matches 9


Number of matches After Lowe's Ratio New 3204
Number of Robust matches New 10
```

```
  2%|▏         | 1/57 [00:00<00:22,  2.44it/s]
```

```
Number of matches 16184
Number of matches After Lowe's Ratio 542
Number of Robust matches 134
```

```
  4%|▎         | 2/57 [00:01<00:28,  1.96it/s]
```

```
Number of matches 24834
Number of matches After Lowe's Ratio 515
Number of Robust matches 6


Number of matches After Lowe's Ratio New 2595
Number of Robust matches New 6
```

```
  5%|▌         | 3/57 [00:02<00:37,  1.43it/s]
```

```
Number of matches 18767
Number of matches After Lowe's Ratio 397
Number of Robust matches 5


Number of matches After Lowe's Ratio New 2136
Number of Robust matches New 8
```

```
  7%|▋         | 4/57 [00:03<00:44,  1.20it/s]
```

```
Number of matches 12135
Number of matches After Lowe's Ratio 587
Number of Robust matches 215
```

```
  9%|▉         | 5/57 [00:03<00:36,  1.41it/s]
```

```
Number of matches 6159
Number of matches After Lowe's Ratio 193
Number of Robust matches 9


Number of matches After Lowe's Ratio New 840
Number of Robust matches New 13
```

```
 11%|█         | 6/57 [00:04<00:29,  1.70it/s]
```

```
Number of matches 5801
Number of matches After Lowe's Ratio 190
Number of Robust matches 12


Number of matches After Lowe's Ratio New 696
Number of Robust matches New 7
```

```
 12%|█▏        | 7/57 [00:04<00:25,  1.94it/s]
```

```
Number of matches 10277
Number of matches After Lowe's Ratio 277
Number of Robust matches 12


Number of matches After Lowe's Ratio New 1167
Number of Robust matches New 18
```

```
 14%|█▍        | 8/57 [00:04<00:24,  1.99it/s]
```

```
Number of matches 13621
Number of matches After Lowe's Ratio 317
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1482
Number of Robust matches New 7
```

```
 16%|█▌        | 9/57 [00:05<00:25,  1.91it/s]
```

```
Number of matches 15668
Number of matches After Lowe's Ratio 337
Number of Robust matches 15


Number of matches After Lowe's Ratio New 1508
Number of Robust matches New 13


 18%|█▊        | 10/57 [00:06<00:25,  1.84it/s]
Number of matches 12996
Number of matches After Lowe's Ratio 271
Number of Robust matches 7


Number of matches After Lowe's Ratio New 1411
Number of Robust matches New 8


 19%|█▉        | 11/57 [00:07<00:34,  1.33it/s]
Number of matches 29441
Number of matches After Lowe's Ratio 478
Number of Robust matches 6


Number of matches After Lowe's Ratio New 2774
Number of Robust matches New 7



Number of matches 95635
Number of matches After Lowe's Ratio 1547
Number of Robust matches 7


Number of matches After Lowe's Ratio New 9451
 21%|██        | 12/57 [00:11<01:17,  1.73s/it]
Number of Robust matches New 8



Number of matches 99466
Number of matches After Lowe's Ratio 1334
Number of Robust matches 5


Number of matches After Lowe's Ratio New 8291
 23%|██▎       | 13/57 [00:19<02:38,  3.61s/it]
Number of Robust matches New 6


 25%|██▍       | 14/57 [00:28<03:40,  5.14s/it]
Number of matches 118808
Number of matches After Lowe's Ratio 2541
Number of Robust matches 191



Number of matches 72817
Number of matches After Lowe's Ratio 1393
Number of Robust matches 6


Number of matches After Lowe's Ratio New 7287
```

```
 26%|██▋       | 15/57 [00:34<03:57,  5.65s/it]
Number of Robust matches New 7



Number of matches 108020
Number of matches After Lowe's Ratio 1619
Number of Robust matches 5


Number of matches After Lowe's Ratio New 9246
 28%|██▊       | 16/57 [00:43<04:28,  6.55s/it]
Number of Robust matches New 7



Number of matches 97178
Number of matches After Lowe's Ratio 1496
Number of Robust matches 6


Number of matches After Lowe's Ratio New 8776
 30%|██▉       | 17/57 [00:51<04:37,  6.94s/it]
Number of Robust matches New 7



Number of matches 119943
Number of matches After Lowe's Ratio 2189
Number of Robust matches 6


Number of matches After Lowe's Ratio New 11773
 32%|███▏      | 18/57 [01:00<04:51,  7.48s/it]
Number of Robust matches New 6



Number of matches 107903
Number of matches After Lowe's Ratio 2256
Number of Robust matches 5


Number of matches After Lowe's Ratio New 11495
 33%|███▎      | 19/57 [01:10<05:11,  8.21s/it]
Number of Robust matches New 6


 35%|███▌      | 20/57 [01:17<04:57,  8.04s/it]
Number of matches 97901
Number of matches After Lowe's Ratio 2446
Number of Robust matches 232



Number of matches 91634
Number of matches After Lowe's Ratio 1395
Number of Robust matches 6


Number of matches After Lowe's Ratio New 8451
 37%|███▋      | 21/57 [01:25<04:41,  7.82s/it]
```

```
Number of Robust matches New 6



Number of matches 87700
Number of matches After Lowe's Ratio 1464
Number of Robust matches 6


Number of matches After Lowe's Ratio New 8092
```

```
 39%|███▉      | 22/57 [01:32<04:27,  7.64s/it]
```

```
Number of Robust matches New 6
```

```
 40%|████      | 23/57 [01:38<04:01,  7.10s/it]
```

```
Number of matches 65556
Number of matches After Lowe's Ratio 1573
Number of Robust matches 326
```

```
 42%|████▏     | 24/57 [01:42<03:24,  6.19s/it]
```

```
Number of matches 43335
Number of matches After Lowe's Ratio 1860
Number of Robust matches 668
```

```
 44%|████▍     | 25/57 [01:43<02:32,  4.76s/it]
```

```
Number of matches 6541
Number of matches After Lowe's Ratio 186
Number of Robust matches 9


Number of matches After Lowe's Ratio New 842
Number of Robust matches New 9
```

```
 46%|████▌     | 26/57 [01:44<01:47,  3.46s/it]
```

```
Number of matches 16578
Number of matches After Lowe's Ratio 338
Number of Robust matches 5


Number of matches After Lowe's Ratio New 1663
Number of Robust matches New 10
```

```
 47%|████▋     | 27/57 [01:45<01:24,  2.83s/it]
```

```
Number of matches 19531
Number of matches After Lowe's Ratio 522
Number of Robust matches 8


Number of matches After Lowe's Ratio New 2386
Number of Robust matches New 9
```

```
 49%|████▉     | 28/57 [01:46<01:04,  2.23s/it]
```

```
Number of matches 14965
Number of matches After Lowe's Ratio 429
Number of Robust matches 7
Number of matches After Lowe's Ratio New 1931
Number of Robust matches New 16
```

```
 51%|█████     | 29/57 [01:46<00:48,  1.72s/it]
```

```
Number of matches 8747
Number of matches After Lowe's Ratio 221
Number of Robust matches 6


Number of matches After Lowe's Ratio New 1029
Number of Robust matches New 8
```

```
 53%|█████▎    | 30/57 [01:47<00:35,  1.31s/it]
```

```
Number of matches 5191
Number of matches After Lowe's Ratio 167
Number of Robust matches 6


Number of matches After Lowe's Ratio New 674
Number of Robust matches New 11
```

```
 54%|█████▍    | 31/57 [01:47<00:25,  1.00it/s]
```

```
Number of matches 10542
Number of matches After Lowe's Ratio 479
Number of Robust matches 84
```

```
 56%|█████▌    | 32/57 [01:47<00:21,  1.18it/s]
```

```
Number of matches 23590
Number of matches After Lowe's Ratio 658
Number of Robust matches 81



Number of matches 32596
Number of matches After Lowe's Ratio 678
Number of Robust matches 5


Number of matches After Lowe's Ratio New 3576
```

```
 58%|█████▊    | 33/57 [01:49<00:28,  1.19s/it]
```

```
Number of Robust matches New 6
```

```
 60%|█████▉    | 34/57 [01:51<00:31,  1.37s/it]
```

```
Number of matches 25114
Number of matches After Lowe's Ratio 546
Number of Robust matches 7


Number of matches After Lowe's Ratio New 2804
Number of Robust matches New 8



Number of matches 33245
Number of matches After Lowe's Ratio 713
Number of Robust matches 6
Number of matches After Lowe's Ratio New 3662
Number of Robust matches New

 61%|██████    | 35/57 [01:53<00:31,  1.42s/it]

 9



 63%|██████▎   | 36/57 [01:55<00:37,  1.79s/it]

Number of matches 31588
Number of matches After Lowe's Ratio 587
Number of Robust matches 6


Number of matches After Lowe's Ratio New 3326
Number of Robust matches New 15



Number of matches 32989
Number of matches After Lowe's Ratio 716
Number of Robust matches 5


Number of matches After Lowe's Ratio New 3757

 65%|██████▌   | 37/57 [01:57<00:37,  1.90s/it]

Number of Robust matches New 7



 67%|██████▋   | 38/57 [01:59<00:35,  1.89s/it]

Number of matches 25202
Number of matches After Lowe's Ratio 566
Number of Robust matches 6


Number of matches After Lowe's Ratio New 2916
Number of Robust matches New 7



 68%|██████▊   | 39/57 [02:01<00:33,  1.85s/it]

Number of matches 21835
Number of matches After Lowe's Ratio 547
Number of Robust matches 5


Number of matches After Lowe's Ratio New 2585
Number of Robust matches New 5



 70%|███████   | 40/57 [02:02<00:26,  1.54s/it]

Number of matches 9061
Number of matches After Lowe's Ratio 220
Number of Robust matches 7


Number of matches After Lowe's Ratio New 1016
Number of Robust matches New 10



 72%|███████▏  | 41/57 [02:03<00:20,  1.26s/it]

Number of matches 26281
Number of matches 20261
Number of matches After Lowe's Ratio 448
Number of Robust matches 6


Number of matches After Lowe's Ratio New 2571
Number of Robust matches New 10


 74%|███████▍  | 42/57 [02:04<00:20,  1.34s/it]
Number of matches 34343
Number of matches After Lowe's Ratio 827
Number of Robust matches 110



Number of matches 35128
Number of matches After Lowe's Ratio 624
Number of Robust matches 5


Number of matches After Lowe's Ratio New 3630
 75%|███████▌  | 43/57 [02:07<00:25,  1.79s/it]
Number of Robust matches New 9



Number of matches 67968
Number of matches After Lowe's Ratio 1036
Number of Robust matches 6


Number of matches After Lowe's Ratio New 6557
 77%|███████▋  | 44/57 [02:11<00:30,  2.38s/it]
Number of Robust matches New 9



Number of matches 91806
Number of matches After Lowe's Ratio 1195
Number of Robust matches 5


Number of matches After Lowe's Ratio New 7715
 79%|███████▉  | 45/57 [02:19<00:48,  4.05s/it]
Number of Robust matches New 6



Number of matches 81079
Number of matches After Lowe's Ratio 1344
Number of Robust matches 6


Number of matches After Lowe's Ratio New 7737
 81%|████████  | 46/57 [02:26<00:56,  5.17s/it]
Number of Robust matches New 6



Number of matches 111456
Number of matches After Lowe's Ratio 1771
Number of Robust matches 5
Number of matches After Lowe's Ratio New 10076
```

```
 82%|████████▏ | 47/57 [02:36<01:03,  6.37s/it]
```

```
Number of Robust matches New 6



Number of matches 74626
Number of matches After Lowe's Ratio 904
Number of Robust matches 5


Number of matches After Lowe's Ratio New 6188
```

```
 84%|████████▍ | 48/57 [02:42<00:58,  6.52s/it]
```

```
Number of Robust matches New 6
```

```
 86%|████████▌ | 49/57 [02:49<00:51,  6.43s/it]
```

```
Number of matches 73113
Number of matches After Lowe's Ratio 2066
Number of Robust matches 496
```

```
 88%|████████▊ | 50/57 [02:54<00:41,  5.96s/it]
```

```
Number of matches 61953
Number of matches After Lowe's Ratio 1807
Number of Robust matches 426
```

```
 89%|████████▉ | 51/57 [02:59<00:34,  5.70s/it]
```

```
Number of matches 54791
Number of matches After Lowe's Ratio 1411
Number of Robust matches 274
```

```
 91%|█████████ | 52/57 [03:02<00:24,  4.87s/it]
```

```
Number of matches 30398
Number of matches After Lowe's Ratio 751
Number of Robust matches 119



Number of matches 31781
Number of matches After Lowe's Ratio 702
Number of Robust matches 7


Number of matches After Lowe's Ratio New 3540
```

```
 93%|█████████▎| 53/57 [03:04<00:16,  4.08s/it]
```

```
Number of Robust matches New 10
```

```
 95%|█████████▍| 54/57 [03:06<00:10,  3.39s/it]
```

```
Number of matches 35560
Number of matches After Lowe's Ratio 1606
Number of Robust matches 612
```

```
 96%|█████████▋| 55/57 [03:07<00:05,  2.84s/it]
```

```
Number of matches 18768
Number of matches After Lowe's Ratio 347
Number of Robust matches 35



Number of matches 33504
Number of matches After Lowe's Ratio 755
Number of Robust matches 6


Number of matches After Lowe's Ratio New 3878
```

```
 98%|█████████▊| 56/57 [03:09<00:02,  2.51s/it]
```

```
Number of Robust matches New 6
```

In [89]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_brief_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_brief)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_brief_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.008490562438964844 [s] ... size 0.00608 MB
```

In [90]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_brief_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_brief)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_r
ight_brief_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.0022048950195 3125 [s] ... size 0.00608 MB
```

In [91]:

```
del H_left_brief, H_right_brief,keypoints_all_left_star, keypoints_all_right_star, descri
ptors_all_left_brief, descriptors_all_right_brief, points_all_left_star, points_all_right
_star
```

In [ ]:

```
import pickle
Fdb = open('all_feat_agast_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_agast = []
descriptors_all_left_agast = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_agast.append(keypoints_each)
  descriptors_all_left_agast.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_agast_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_agast = []
descriptors_all_right_agast = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each])
)
  keypoints_all_right_agast.append(keypoints_each)
  descriptors_all_right_agast.append(descrip_each)
```

In [ ]:

```
H_left_agast = []
H_right_agast = []

num_matches_agast = []
num_good_matches_agast = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_a
gast[j:j+2][::-1],points_all_left_agast[j:j+2][::-1],descriptors_all_left_agast[j:j+2][:
:-1],0.85,6)
  H_left_agast.append(H_a)
  num_matches_agast.append(matches)
  num_good_matches_agast.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_agast[j:j+2][::-1],points_all_right_agast[j:j+2][::-1],descriptors_all_right_agast[j:j+
2][::-1],0.85,6)
  H_right_agast.append(H_a)
  num_matches_agast.append(matches)
  num_good_matches_agast.append(gd_matches)
```

In [ ]:

```
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_agast_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_agast)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_agast_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_agast_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_agast)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_r
ight_agast_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_agast, H_right_agast,keypoints_all_left_agast, keypoints_all_right_agast, desc
riptors_all_left_agast, descriptors_all_right_agast, points_all_left_agast, points_all_ri
ght_agast
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```
import pickle
Fdb = open('all_feat_daisy_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_daisy = []
descriptors_all_left_daisy = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_daisy.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_daisy.append(keypoints_each)
  descriptors_all_left_daisy.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_daisy_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_daisy = []
descriptors_all_right_daisy = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_daisy.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each])
)
  keypoints_all_right_daisy.append(keypoints_each)
  descriptors_all_right_daisy.append(descrip_each)
```

In [ ]:

```
H_left_daisy = []
H_right_daisy = []

num_matches_daisy = []
num_good_matches_daisy = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_d
aisy[j:j+2][::-1],points_all_left_daisy[j:j+2][::-1],descriptors_all_left_daisy[j:j+2][:
:-1],0.7,6)
  H_left_daisy.append(H_a)
  num_matches_daisy.append(matches)
  num_good_matches_daisy.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_daisy[j:j+2][::-1],points_all_right_daisy[j:j+2][::-1],descriptors_all_right_daisy[j:j+
2][::-1],0.7,6)
  H_right_daisy.append(H_a)
  num_matches_daisy.append(matches)
  num_good_matches_daisy.append(gd_matches)
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_daisy_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_daisy)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_daisy_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_daisy_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_daisy)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_r
ight_daisy_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_daisy, H_right_daisy,keypoints_all_left_daisy, keypoints_all_right_daisy, desc
riptors_all_left_daisy, descriptors_all_right_daisy, points_all_left_daisy, points_all_ri
ght_daisy
```

In [ ]:

```
import pickle
Fdb = open('all_feat_freak_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_freak = []
```

```
descriptors_all_left_freak = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_freak.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_freak.append(keypoints_each)
  descriptors_all_left_freak.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_freak_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_freak = []
descriptors_all_right_freak = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_freak.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each])
)
  keypoints_all_right_freak.append(keypoints_each)
  descriptors_all_right_freak.append(descrip_each)
```

In [ ]:

```
H_left_freak = []
H_right_freak = []

num_matches_freak = []
num_good_matches_freak = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_f
reak[j:j+2][::-1],points_all_left_freak[j:j+2][::-1],descriptors_all_left_freak[j:j+2][:
:-1],0.7,6)
  H_left_freak.append(H_a)
  num_matches_freak.append(matches)
  num_good_matches_freak.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_freak[j:j+2][::-1],points_all_right_freak[j:j+2][::-1],descriptors_all_right_freak[j:j+
2][::-1],0.7,6)
  H_right_freak.append(H_a)
  num_matches_freak.append(matches)
  num_good_matches_freak.append(gd_matches)
```

In [ ]:

```

```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_freak_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_freak)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_freak_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_freak_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_freak)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_freak_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_freak, H_right_freak,keypoints_all_left_freak, keypoints_all_right_freak, descriptors_all_left_freak, descriptors_all_right_freak, points_all_left_freak, points_all_right_freak
```

In [ ]:

```
import pickle
Fdb = open('all_feat_surf_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_surf = []
descriptors_all_left_surf = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_surf.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_surf.append(keypoints_each)
  descriptors_all_left_surf.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_surf_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_surf = []
descriptors_all_right_surf = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_surf.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_surf.append(keypoints_each)
  descriptors_all_right_surf.append(descrip_each)
```

In [ ]:

```
H_left_surf = []
H_right_surf = []

num_matches_surf = []
num_good_matches_surf = []

for j in tqdm(range(len(left_files_path))):
  if j==len(left_files_path)-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],0.65)
  H_left_surf.append(H_a)
  num_matches_surf.append(matches)
  num_good_matches_surf.append(gd_matches)

for j in tqdm(range(len(right_files_path))):
  if j==len(right_files_path)-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],0.65)
  H_right_surf.append(H_a)
  num_matches_surf.append(matches)
  num_good_matches_surf.append(gd_matches)
```

In [ ]:

```

```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_surf_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_surf)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_surf_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_surf_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_surf)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_surf_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_surf, H_right_surf,keypoints_all_left_surf, keypoints_all_right_surf, descriptors_all_left_surf, descriptors_all_right_surf, points_all_left_surf, points_all_right_surf
```

In [ ]:

```
import pickle
Fdb = open('all_feat_rootsift_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_rootsift = []
descriptors_all_left_rootsift = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_rootsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each
]))
  keypoints_all_left_rootsift.append(keypoints_each)
  descriptors_all_left_rootsift.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_rootsift_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_rootsift = []
descriptors_all_right_rootsift = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_rootsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_eac
h]))
  keypoints_all_right_rootsift.append(keypoints_each)
  descriptors_all_right_rootsift.append(descrip_each)
```

In [ ]:

```
H_left_rootsift = []
H_right_rootsift = []

num_matches_rootsift = []
num_good_matches_rootsift = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_r
ootsift[j:j+2][::-1],points_all_left_rootsift[j:j+2][::-1],descriptors_all_left_rootsift
[j:j+2][::-1],0.9)
  H_left_rootsift.append(H_a)
  num_matches_rootsift.append(matches)
  num_good_matches_rootsift.append(gd_matches)
```

```
for j in tqdm(range(len(right_files_path))):
  if j==len(right_files_path)-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],0.9)
  H_right_rootsift.append(H_a)
  num_matches_rootsift.append(matches)
  num_good_matches_rootsift.append(gd_matches)
```

In [ ]:

```

```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_rootsift_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_rootsift)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_rootsift_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_rootsift_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_rootsift)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_rootsift_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_rootsift, H_right_rootsift,keypoints_all_left_rootsift, keypoints_all_right_rootsift, descriptors_all_left_rootsift, descriptors_all_right_rootsift, points_all_left_rootsift, points_all_right_rootsift
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```
'''
import pickle
Fdb = open('all_feat_surfsift_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_surfsift = []
descriptors_all_left_surfsift = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                                _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]
))
  keypoints_all_left_surfsift.append(keypoints_each)
  descriptors_all_left_surfsift.append(descrip_each)
'''
```

In [ ]:

```
'''
import pickle
Fdb = open('all_feat_surfsift_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_surfsift = []
descriptors_all_right_surfsift = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=
kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each
]))
  keypoints_all_right_surfsift.append(keypoints_each)
  descriptors_all_right_surfsift.append(descrip_each)
'''
```

In [ ]:

```
'''
H_left_surfsift = []
H_right_surfsift = []

num_matches_surfsift = []
num_good_matches_surfsift = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_su
rfsift[j:j+2][::-1],points_all_left_surfsift[j:j+2][::-1],descriptors_all_left_surfsift[j
:j+2][::-1],0.7,6)
  H_left_surfsift.append(H_a)
  num_matches_surfsift.append(matches)
  num_good_matches_surfsift.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_
surfsift[j:j+2][::-1],points_all_right_surfsift[j:j+2][::-1],descriptors_all_right_surfsi
ft[j:j+2][::-1],0.7,6)
  H_right_surfsift.append(H_a)
  num_matches_surfsift.append(matches)
  num_good_matches_surfsift.append(gd_matches)
'''
```

In [ ]:

```
'''
```

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_surfsift_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_surfsift)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_surfsift_40.h5')/1.e6,'MB')
'''
```

In [ ]:

```
'''
import h5py as h5
f=h5.File('drive/MyDrive/H_right_surfsift_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_surfsift)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_r
ight_surfsift_40.h5')/1.e6,'MB')
'''
```

In [ ]:

```
#del H_left_surfsift, H_right_surfsift,keypoints_all_left_surfsift, keypoints_all_right_s
urfsift, descriptors_all_left_surfsift, descriptors_all_right_surfsift, points_all_left_s
urfsift, points_all_right_surfsift
```

In [ ]:

```

```

In [ ]:

```

```

In [92]:

```
import pickle
Fdb = open('all_feat_gftt_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_gftt = []
descriptors_all_left_gftt = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_gftt.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_gftt.append(keypoints_each)
  descriptors_all_left_gftt.append(descrip_each)
```

In [93]:

```
import pickle
Fdb = open('all_feat_gftt_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_gftt = []
descriptors_all_right_gftt = []
```

```
for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_gftt.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_gftt.append(keypoints_each)
  descriptors_all_right_gftt.append(descrip_each)
```

In [94]:

```
H_left_gftt = []
H_right_gftt = []

num_matches_gftt = []
num_good_matches_gftt = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_g
ftt[j:j+2][::-1],points_all_left_gftt[j:j+2][::-1],descriptors_all_left_gftt[j:j+2][::-1
],0.85,6)
  H_left_gftt.append(H_a)
  num_matches_gftt.append(matches)
  num_good_matches_gftt.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_gftt[j:j+2][::-1],points_all_right_gftt[j:j+2][::-1],descriptors_all_right_gftt[j:j+2][
::-1],0.85,6)
  H_right_gftt.append(H_a)
  num_matches_gftt.append(matches)
  num_good_matches_gftt.append(gd_matches)
```

```
  2%|▏         | 1/57 [00:00<00:11,  4.90it/s]
```

```
Number of matches 1000
Number of matches After Lowe's Ratio 46
Number of Robust matches 8


Number of matches After Lowe's Ratio New 46
Number of Robust matches New 8



Number of matches 1000
Number of matches After Lowe's Ratio 57
```

```
  4%|▍         | 2/57 [00:00<00:11,  4.95it/s]
```

```
Number of Robust matches 7


Number of matches After Lowe's Ratio New 57
Number of Robust matches New 7



Number of matches 1000
Number of matches After Lowe's Ratio 100
Number of Robust matches 12
Number of Robust matches 12


Number of matches After Lowe's Ratio New 100
  5%|▌         | 3/57 [00:00<00:10,  4.96it/s]
Number of Robust matches New 11



Number of matches 1000
Number of matches After Lowe's Ratio 94
  7%|▋         | 4/57 [00:00<00:10,  4.92it/s]
Number of Robust matches 6


Number of matches After Lowe's Ratio New 94
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 144
  9%|▉         | 5/57 [00:01<00:10,  4.88it/s]
Number of Robust matches 24


Number of matches After Lowe's Ratio New 144
Number of Robust matches New 22



Number of matches 1000
Number of matches After Lowe's Ratio 53
Number of Robust matches 15


Number of matches After Lowe's Ratio New 53
 12%|█▏        | 7/57 [00:01<00:09,  5.25it/s]
Number of Robust matches New 12



Number of matches 1000
Number of matches After Lowe's Ratio 60
Number of Robust matches 11


Number of matches After Lowe's Ratio New 60
Number of Robust matches New 10


 16%|█▌        | 9/57 [00:01<00:07,  6.16it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 86
Number of Robust matches 31



Number of matches 1000
Number of matches After Lowe's Ratio 55
Number of Robust matches 17


Number of matches After Lowe's Ratio New 55
Number of Robust matches New 15
Number of Robust matches New 15



Number of matches 1000
Number of matches After Lowe's Ratio 71
Number of Robust matches 26


 19%|█▉        | 11/57 [00:01<00:07,  6.45it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 53
Number of Robust matches 5


Number of matches After Lowe's Ratio New 53
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 85
 21%|██        | 12/57 [00:02<00:07,  5.93it/s]
Number of Robust matches 18


Number of matches After Lowe's Ratio New 85
Number of Robust matches New 18



Number of matches 1000
Number of matches After Lowe's Ratio 55
 23%|██▎       | 13/57 [00:02<00:07,  5.62it/s]
Number of Robust matches 6


Number of matches After Lowe's Ratio New 55
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 58
Number of Robust matches 9
 25%|██▌       | 14/57 [00:02<00:07,  5.38it/s]

Number of matches After Lowe's Ratio New 58
Number of Robust matches New 9



Number of matches 1000
Number of matches After Lowe's Ratio 46
Number of Robust matches
 26%|██▋       | 15/57 [00:02<00:07,  5.28it/s]
 6


Number of matches After Lowe's Ratio New 46
Number of Robust matches New 6



Number of matches 1000
Number of matches 1000
Number of matches After Lowe's Ratio 106
Number of Robust matches 88
```

```
 30%|██▉       | 17/57 [00:02<00:06,  5.71it/s]
```

```
Number of matches 1000
Number of matches After Lowe's Ratio 30
Number of Robust matches 5


Number of matches After Lowe's Ratio New 30
Number of Robust matches New 4
```

```
 32%|███▏      | 18/57 [00:03<00:07,  5.35it/s]
```

```
Number of matches 1000
Number of matches After Lowe's Ratio 41
Number of Robust matches 5


Number of matches After Lowe's Ratio New 41
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 39
```

```
 33%|███▎      | 19/57 [00:03<00:07,  5.24it/s]
```

```
Number of Robust matches 5


Number of matches After Lowe's Ratio New 39
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 37
Number of Robust matches 5
```

```
 35%|███▌      | 20/57 [00:03<00:07,  5.17it/s]
```

```
Number of matches After Lowe's Ratio New 37
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 46
Number of Robust matches 6


Number of matches After Lowe's Ratio New 46
```

```
 37%|███▋      | 21/57 [00:03<00:06,  5.19it/s]
```

```
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 72
Number of Robust matches 8


Number of matches After Lowe's Ratio New 72
Number of Robust matches New 8
Number of Robust matches New 6


 40%|████      | 23/57 [00:04<00:06,  5.09it/s]

Number of matches 1000
Number of matches After Lowe's Ratio 56
Number of Robust matches 6


Number of matches After Lowe's Ratio New 56
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 114

 42%|████▏     | 24/57 [00:04<00:06,  5.08it/s]

Number of Robust matches 15


Number of matches After Lowe's Ratio New 114
Number of Robust matches New 15



Number of matches 1000
Number of matches After Lowe's Ratio 50
Number of Robust matches 12


Number of matches After Lowe's Ratio New 50

 46%|████▌     | 26/57 [00:04<00:05,  5.38it/s]

Number of Robust matches New 11



Number of matches 1000
Number of matches After Lowe's Ratio 40
Number of Robust matches 9


Number of matches After Lowe's Ratio New 40
Number of Robust matches New 8



Number of matches 1000
Number of matches After Lowe's Ratio 43
Number of Robust matches 15


Number of matches After Lowe's Ratio New 43
Number of Robust matches New 13



Number of matches 1000
Number of matches After Lowe's Ratio 50
Number of Robust matches 9


Number of matches After Lowe's Ratio New 50

 49%|████▉     | 28/57 [00:04<00:05,  5.76it/s]

Number of Robust matches New 9
Number of matches 1000
Number of matches After Lowe's Ratio 87
Number of Robust matches 13


Number of matches After Lowe's Ratio New 87
 54%|█████     | 31/57 [00:05<00:04,  6.47it/s]
Number of Robust matches New 12



Number of matches 1000
Number of matches After Lowe's Ratio 101
Number of Robust matches 38



Number of matches 1000
Number of matches After Lowe's Ratio 56
Number of Robust matches 23


Number of matches After Lowe's Ratio New 56
Number of Robust matches New 24



Number of matches 1000
Number of matches After Lowe's Ratio 23
Number of Robust matches 10


Number of matches After Lowe's Ratio New 23
Number of Robust matches New 10



Number of matches 1000
Number of matches After Lowe's Ratio 55
Number of Robust matches 11


Number of matches After Lowe's Ratio New 55
 60%|██████    | 34/57 [00:05<00:03,  6.16it/s]
Number of Robust matches New 11



Number of matches 1000
Number of matches After Lowe's Ratio 100
Number of Robust matches 21


Number of matches After Lowe's Ratio New 100
Number of Robust matches New 19


 61%|██████    | 35/57 [00:05<00:03,  5.79it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 104
Number of Robust matches 15


Number of matches After Lowe's Ratio New 104
Number of Robust matches New 15
Number of matches 1000
Number of matches After Lowe's Ratio 45
 63%|██████▎   | 36/57 [00:06<00:03,  6.20it/s]
Number of Robust matches 11


Number of matches After Lowe's Ratio New 45
Number of Robust matches New 11



Number of matches 1000
Number of matches After Lowe's Ratio 62
Number of Robust matches 7


Number of matches After Lowe's Ratio New 62
 65%|██████▍   | 37/57 [00:06<00:03,  5.75it/s]
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 53
Number of Robust matches 9


Number of matches After Lowe's Ratio New 53
 67%|██████▋   | 38/57 [00:06<00:03,  5.46it/s]
Number of Robust matches New 7



Number of matches 1000
Number of matches After Lowe's Ratio 76
Number of Robust matches 32



Number of matches 1000
Number of matches After Lowe's Ratio 43
 72%|███████▏  | 41/57 [00:06<00:02,  6.70it/s]
Number of Robust matches 5


Number of matches After Lowe's Ratio New 43
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 27
Number of Robust matches 8


Number of matches After Lowe's Ratio New 27
Number of Robust matches New 8



Number of matches 1000
Number of matches After Lowe's Ratio 250
Number of Robust matches 187
Number of matches 1000
Number of matches After Lowe's Ratio 42
Number of Robust matches 6


Number of matches After Lowe's Ratio New 42
 77%|███████▋  | 44/57 [00:07<00:02,  6.16it/s]
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 43
Number of Robust matches 5


Number of matches After Lowe's Ratio New 43
Number of Robust matches New 5


 81%|████████  | 46/57 [00:07<00:01,  6.48it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 122
Number of Robust matches 73



Number of matches 1000
Number of matches After Lowe's Ratio 44
Number of Robust matches 8


Number of matches After Lowe's Ratio New 44
Number of Robust matches New 7


 82%|████████▏ | 47/57 [00:07<00:01,  6.00it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 54
Number of Robust matches 10


Number of matches After Lowe's Ratio New 54
Number of Robust matches New 10



Number of matches 1000
Number of matches After Lowe's Ratio 67
Number of Robust matches 36


 86%|████████▌ | 49/57 [00:08<00:01,  6.38it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 54
Number of Robust matches 10


Number of matches After Lowe's Ratio New 54
Number of Robust matches New 10



Number of matches 1000
Number of matches 1000
Number of matches After Lowe's Ratio 155

 88%|████████▊ | 50/57 [00:08<00:01,  6.65it/s]

Number of Robust matches 27



Number of matches 1000
Number of matches After Lowe's Ratio 83
Number of Robust matches 29



Number of matches 1000
Number of matches After Lowe's Ratio 66

 91%|█████████ | 52/57 [00:08<00:00,  6.66it/s]

Number of Robust matches 13


Number of matches After Lowe's Ratio New 66
Number of Robust matches New 13



Number of matches 1000
Number of matches After Lowe's Ratio 71

 93%|█████████▎| 53/57 [00:08<00:00,  6.02it/s]

Number of Robust matches 6


Number of matches After Lowe's Ratio New 71
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 75
Number of Robust matches 11


Number of matches After Lowe's Ratio New 75

  0%|          | 0/57 [00:00<?, ?it/s]

Number of Robust matches New 9



Number of matches 1000
Number of matches After Lowe's Ratio 44
Number of Robust matches 17


Number of matches After Lowe's Ratio New 44
Number of Robust matches New 17



Number of matches 1000
Number of matches After Lowe's Ratio 77
Number of Robust matches 22


Number of matches After Lowe's Ratio New 77
Number of Robust matches New 21



  2%|▏         | 1/57 [00:00<00:11,  4.87it/s]
  2%|▏         | 1/57 [00:00<00:11,  4.87it/s]
```

```
Number of matches 1000
Number of matches After Lowe's Ratio 95
Number of Robust matches 12


Number of matches After Lowe's Ratio New 95
Number of Robust matches New 12



Number of matches 1000
Number of matches After Lowe's Ratio 128
Number of Robust matches 39



Number of matches 1000
Number of matches After Lowe's Ratio 65
Number of Robust matches 10


Number of matches After Lowe's Ratio New 65
```

```
  5%|▌         | 3/57 [00:00<00:10,  5.29it/s]
```

```
Number of Robust matches New 10



Number of matches 1000
Number of matches After Lowe's Ratio 138
Number of Robust matches 43



Number of matches 1000
Number of matches After Lowe's Ratio 78
```

```
  9%|▊         | 5/57 [00:00<00:09,  5.69it/s]
```

```
Number of Robust matches 11


Number of matches After Lowe's Ratio New 78
Number of Robust matches New 10



Number of matches 1000
Number of matches After Lowe's Ratio 98
```

```
 11%|█         | 6/57 [00:01<00:09,  5.40it/s]
```

```
Number of Robust matches 13


Number of matches After Lowe's Ratio New 98
Number of Robust matches New 13



Number of matches 1000
Number of matches After Lowe's Ratio 115
Number of Robust matches 9


Number of matches After Lowe's Ratio New 115
```

```
 12%|█▏        | 7/57 [00:01<00:09,  5.29it/s]
```

```
Number of Robust matches New 8
Number of matches 1000
Number of matches After Lowe's Ratio 113
Number of Robust matches 32



Number of matches 1000
Number of matches After Lowe's Ratio 50
```

```
 16%|█▌        | 9/57 [00:01<00:08,  5.70it/s]
```

```
Number of Robust matches 6


Number of matches After Lowe's Ratio New 50
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 42
```

```
 18%|█▊        | 10/57 [00:01<00:08,  5.32it/s]
```

```
Number of Robust matches 5


Number of matches After Lowe's Ratio New 42
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 46
Number of Robust matches 6


Number of matches After Lowe's Ratio New 46
```

```
 21%|██        | 12/57 [00:02<00:08,  5.18it/s]
```

```
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 48
Number of Robust matches 9


Number of matches After Lowe's Ratio New 48
Number of Robust matches New 8
```

```
 23%|██▎       | 13/57 [00:02<00:08,  5.16it/s]
```

```
Number of matches 1000
Number of matches After Lowe's Ratio 50
Number of Robust matches 5


Number of matches After Lowe's Ratio New 50
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 44
```

```
 25%|██▌       | 14/57 [00:02<00:08,  5.14it/s]
```

```
Number of Robust matches 5


Number of matches After Lowe's Ratio New 44
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 40
Number of Robust matches 5


Number of matches After Lowe's Ratio New 40
```

```
 28%|██▊       | 16/57 [00:02<00:07,  5.14it/s]
```

```
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 34
Number of Robust matches 5


Number of matches After Lowe's Ratio New 34
Number of Robust matches New 5
```

```
 30%|██▉       | 17/57 [00:03<00:07,  5.03it/s]
```

```
Number of matches 1000
Number of matches After Lowe's Ratio 39
Number of Robust matches 5


Number of matches After Lowe's Ratio New 39
Number of Robust matches New 5
```

```
 32%|███▏      | 18/57 [00:03<00:07,  5.04it/s]
```

```
Number of matches 1000
Number of matches After Lowe's Ratio 41
Number of Robust matches 5


Number of matches After Lowe's Ratio New 41
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 40
```

```
 33%|███▎      | 19/57 [00:03<00:07,  5.02it/s]
```

```
Number of Robust matches 5


Number of matches After Lowe's Ratio New 40
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 92
Number of Robust matches 51
```

```
 37%|███▋      | 21/57 [00:03<00:06,  5.58it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 47
Number of Robust matches 5


Number of matches After Lowe's Ratio New 47
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 56
 40%|████      | 23/57 [00:03<00:04,  6.88it/s]
Number of Robust matches 16


Number of matches After Lowe's Ratio New 56
Number of Robust matches New 14



Number of matches 1000
Number of matches After Lowe's Ratio 73
Number of Robust matches 21


Number of matches After Lowe's Ratio New 73
Number of Robust matches New 21


 44%|████▍     | 25/57 [00:04<00:03,  8.05it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 129
Number of Robust matches 67



Number of matches 1000
Number of matches After Lowe's Ratio 135
Number of Robust matches 52



Number of matches 1000
Number of matches After Lowe's Ratio 112
Number of Robust matches 27


 47%|████▋     | 27/57 [00:04<00:03,  7.61it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 110
Number of Robust matches 15


Number of matches After Lowe's Ratio New 110
Number of Robust matches New 12



Number of matches 1000
Number of matches After Lowe's Ratio 133
 49%|████▉     | 28/57 [00:04<00:04,  6.61it/s]
Number of Robust matches 17
Number of matches After Lowe's Ratio New 133
Number of Robust matches New 17



Number of matches 1000
Number of matches After Lowe's Ratio 55

 51%|█████     | 29/57 [00:04<00:04,  5.92it/s]

Number of Robust matches 7


Number of matches After Lowe's Ratio New 55
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 90

 53%|█████▎    | 30/57 [00:05<00:04,  5.57it/s]

Number of Robust matches 13


Number of matches After Lowe's Ratio New 90
Number of Robust matches New 13



Number of matches 1000
Number of matches After Lowe's Ratio 120

 54%|█████▍    | 31/57 [00:05<00:04,  5.35it/s]

Number of Robust matches 20


Number of matches After Lowe's Ratio New 120
Number of Robust matches New 14



Number of matches 1000
Number of matches After Lowe's Ratio 139
Number of Robust matches 61



Number of matches 1000
Number of matches After Lowe's Ratio 60

 58%|█████▊    | 33/57 [00:05<00:04,  5.84it/s]

Number of Robust matches 6


Number of matches After Lowe's Ratio New 60
Number of Robust matches New 6



Number of matches 1000
Number of matches After Lowe's Ratio 89

 60%|█████▉    | 34/57 [00:05<00:04,  5.54it/s]

Number of Robust matches 16


Number of matches After Lowe's Ratio New 89
Number of Robust matches New 12
Number of matches 1000
Number of matches After Lowe's Ratio 49
Number of Robust matches 7


Number of matches After Lowe's Ratio New 49
 61%|██████    | 35/57 [00:05<00:04,  5.36it/s]
Number of Robust matches New 7



Number of matches 1000
Number of matches After Lowe's Ratio 42
Number of Robust matches 5


Number of matches After Lowe's Ratio New 42
 65%|██████▍   | 37/57 [00:06<00:03,  5.21it/s]
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 38
Number of Robust matches 5


Number of matches After Lowe's Ratio New 38
Number of Robust matches New 5


 67%|██████▋   | 38/57 [00:06<00:03,  5.15it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 61
Number of Robust matches 8


Number of matches After Lowe's Ratio New 61
Number of Robust matches New 7



Number of matches 1000
Number of matches After Lowe's Ratio 62
 68%|██████▊   | 39/57 [00:06<00:03,  5.10it/s]
Number of Robust matches 5


Number of matches After Lowe's Ratio New 62
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 58
Number of Robust matches 6


Number of matches After Lowe's Ratio New 58
 72%|███████▏  | 41/57 [00:07<00:03,  5.12it/s]
Number of Robust matches New 5
Number of matches 1000
Number of matches After Lowe's Ratio 52
Number of Robust matches 11


Number of matches After Lowe's Ratio New 52
Number of Robust matches New 10


 75%|███████▌  | 43/57 [00:07<00:02,  6.06it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 79
Number of Robust matches 25



Number of matches 1000
Number of matches After Lowe's Ratio 28
Number of Robust matches 10


Number of matches After Lowe's Ratio New 28
Number of Robust matches New 10



Number of matches 1000
Number of matches After Lowe's Ratio 50
 79%|███████▉  | 45/57 [00:07<00:01,  6.44it/s]
Number of Robust matches 5


Number of matches After Lowe's Ratio New 50
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 54
Number of Robust matches 16


Number of matches After Lowe's Ratio New 54
Number of Robust matches New 15


 81%|████████  | 46/57 [00:07<00:01,  5.97it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 45
Number of Robust matches 5


Number of matches After Lowe's Ratio New 45
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 54
 82%|████████▏ | 47/57 [00:07<00:01,  5.80it/s]
Number of Robust matches 12
Number of matches After Lowe's Ratio New 54
Number of Robust matches New 11



Number of matches 1000
Number of matches After Lowe's Ratio 52
Number of Robust matches 5


Number of matches After Lowe's Ratio New 52
 86%|████████▌ | 49/57 [00:08<00:01,  5.44it/s]
Number of Robust matches New 5



Number of matches 1000
Number of matches After Lowe's Ratio 69
Number of Robust matches 11


Number of matches After Lowe's Ratio New 69
Number of Robust matches New 10


 89%|████████▉ | 51/57 [00:08<00:00,  6.63it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 58
Number of Robust matches 27



Number of matches 1000
Number of matches After Lowe's Ratio 66
Number of Robust matches 32



Number of matches 1000
Number of matches After Lowe's Ratio 92
Number of Robust matches 48


 93%|█████████▎| 53/57 [00:08<00:00,  7.44it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 66
Number of Robust matches 19


Number of matches After Lowe's Ratio New 66
Number of Robust matches New 17



Number of matches 1000
Number of matches After Lowe's Ratio 224
Number of Robust matches 148


 98%|█████████▊| 56/57 [00:09<00:00,  7.05it/s]
Number of matches 1000
Number of matches After Lowe's Ratio 150
Number of Robust matches 108
Number of matches 1000
Number of matches After Lowe's Ratio 38
Number of Robust matches 7


Number of matches After Lowe's Ratio New 38
Number of Robust matches New 7
```

In [95]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_gftt_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_gftt)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_gftt_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.004984855651855469 [s] ... size 0.00608 MB
```

In [96]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_gftt_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_gftt)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_gftt_40.h5')/1.e6,'MB')
```

```
HDF5  w/o comp.: 0.004405975341796875 [s] ... size 0.00608 MB
```

In [97]:

```
del H_left_gftt, H_right_gftt,keypoints_all_left_gftt, keypoints_all_right_gftt, descriptors_all_left_gftt, descriptors_all_right_gftt, points_all_left_gftt, points_all_right_gftt
```

In [ ]:

```
#points_all_left_mser = points_all_right_mser = []
```

In [ ]:

```
import pickle
Fdb = open('all_feat_mser_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_mser = []
descriptors_all_left_mser = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                                _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_mser.append(keypoints_each)
  descriptors_all_left_mser.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_mser_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_mser = []
descriptors_all_right_mser = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle
=kpt_img[2],
                            _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[
5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_mser.append(keypoints_each)
  descriptors_all_right_mser.append(descrip_each)
```

In [ ]:

```
H_left_mser = []
H_right_mser = []

num_matches_mser = []
num_good_matches_mser = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_m
ser[j:j+2][::-1],points_all_left_mser[j:j+2][::-1],descriptors_all_left_mser[j:j+2][::-1
],0.95,8)
  H_left_mser.append(H_a)
  num_matches_mser.append(matches)
  num_good_matches_mser.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right
_mser[j:j+2][::-1],points_all_right_mser[j:j+2][::-1],descriptors_all_right_mser[j:j+2][
::-1],0.95,8)
  H_right_mser.append(H_a)
  num_matches_mser.append(matches)
  num_good_matches_mser.append(gd_matches)
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_mser_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_mser)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_l
eft_mser_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_mser_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_mser)
f.close()
```

```
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_mser_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_mser, H_right_mser,keypoints_all_left_mser, keypoints_all_right_mser, descriptors_all_left_mser, descriptors_all_right_mser, points_all_left_mser, points_all_right_mser
```

In [ ]:

```
import pickle
Fdb = open('all_feat_superpoint_left.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_left_superpoint = []
descriptors_all_left_superpoint = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                                _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_left_superpoint.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_left_superpoint.append(keypoints_each)
  descriptors_all_left_superpoint.append(descrip_each)
```

In [ ]:

```
import pickle
Fdb = open('all_feat_superpoint_right.dat', 'rb')
kpts_all = pickle.load(Fdb)
Fdb.close()

keypoints_all_right_superpoint = []
descriptors_all_right_superpoint = []

for j,kpt_each in enumerate(kpts_all):
  keypoints_each = []
  descrip_each = []
  for k,kpt_img in enumerate(kpt_each):
    temp_feature = cv2.KeyPoint(x=kpt_img[0][0],y=kpt_img[0][1],_size=kpt_img[1], _angle=kpt_img[2],
                                _response=kpt_img[3], _octave=kpt_img[4], _class_id=kpt_img[5])
    temp_descriptor = kpt_img[6]
    keypoints_each.append(temp_feature)
    descrip_each.append(temp_descriptor)
  points_all_right_superpoint.append(np.asarray([[p.pt[0], p.pt[1]] for p in keypoints_each]))
  keypoints_all_right_superpoint.append(keypoints_each)
  descriptors_all_right_superpoint.append(descrip_each)
```

In [ ]:

```
H_left_superpoint = []
H_right_superpoint = []

num_matches_superpoint = []
num_good_matches_superpoint = []

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

  H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left_superpoint[j:j+2][::-1],points_all_left_superpoint[j:j+2][::-1],descriptors_all_left_superpoint[j:j+2][::-1],ratio=0.8,thresh=3,no_ransac=False,use_lowe=True)
  H_left_superpoint.append(H_a)
  num_matches_superpoint.append(matches)
  num_good_matches_superpoint.append(gd_matches)

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

  H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_right_superpoint[j:j+2][::-1],points_all_right_superpoint[j:j+2][::-1],descriptors_all_right_superpoint[j:j+2][::-1],ratio=0.8,thresh = 3,no_ransac=False,use_lowe=True)
  H_right_superpoint.append(H_a)
  num_matches_superpoint.append(matches)
  num_good_matches_superpoint.append(gd_matches)
```

In [ ]:

```

```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_left_superpoint_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_left_superpoint)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_left_superpoint_40.h5')/1.e6,'MB')
```

In [ ]:

```
import h5py as h5
f=h5.File('drive/MyDrive/H_right_superpoint_40.h5','w')
t0=time.time()
f.create_dataset('data',data=H_right_superpoint)
f.close()
print('HDF5  w/o comp.:',time.time()-t0,'[s] ... size',os.path.getsize('drive/MyDrive/H_right_superpoint_40.h5')/1.e6,'MB')
```

In [ ]:

```
del H_left_superpoint, H_right_superpoint,keypoints_all_left_superpoint, keypoints_all_right_superpoint, descriptors_all_left_superpoint, descriptors_all_right_superpoint, points_all_left_superpoint, points_all_right_superpoint
```

In [ ]:

```
print(len(num_matches_superpoint))
```

Evaluation Criteria/Performance Metrics for each Dataset:

- **Total Number of Keypoints/Descriptors** detected for dataset (Higher the better) (Plot for 16 are above) for each detector/descriptor
- **Total Number of Matches** (Higher the better) for each detector/descriptor (Plot for 9 below)
- **Total Number of Good Matches after Lowe ratio and RANSAC** (Higher the better) for each detector/descriptor (Plot for 9 Below)
- **Recall rate** which is the Percentage of Good Matches (Higher the Better) from all total matches b/w corresponding images by each detector/descriptor (Plot for 9 Below)
- **1-Precision rate** which signifies Percentage of False matches (Lower the Better) from each detector/descriptor (Plot for 9 Below)
- **F-Score** which which is the Geometric Mean b/w Recall and Precision rate for matches b/w corresponding images (Higher the Better) from each detector/descriptor (Plot for 9 Below)

- **Time taken by each descriptor/detector (Lower the Better) (Will Plot this after optimization)**

Collect All Number Of KeyPoints

In [ ]:

```
d = {'Dataset': [f'{Dataset}']*(num_detectors*len_files), 'Number of Keypoints':num_kps_
agast+ num_kps_akaze + num_kps_brisk + num_kps_daisy + num_kps_fast + num_kps_freak + num
_kps_gftt + num_kps_kaze + num_kps_mser + num_kps_orb + num_kps_rootsift + num_kps_sift
+ num_kps_briefstar + num_kps_superpoint+ num_kps_surf, 'Detector/Descriptor':['AGAST+SI
FT']*len_files + ['AKAZE']*len_files + ['BRISK']*len_files + ['DAISY+SIFT']*len_files +
['FAST+SIFT']*len_files + ['BRISK+FREAK']*len_files + ['GFTT+SIFT']*len_files + ['KAZE']
*len_files + ['MSER+SIFT']*len_files + ['ORB']*len_files +['RootSIFT']*len_files +['SIFT
']*len_files + ['STAR+BRIEF']*len_files +  ['SuperPoint']*len_files + ['SURF']*len_files
}
df_numkey_15 = pd.DataFrame(data=d)
df_numkey_15['Number of Keypoints'] = df_numkey_15['Number of Keypoints']/(len_files)
```

In [100]:

```
len(left_files_path)
```

Out[100]:

```
57
```

In [107]:

```
len(right_files_path[1:])
```

Out[107]:

```
56
```

In [105]:

```
len_files = len(left_files_path) + len(right_files_path[1:])
num_detectors = 15
```

In [109]:

```
d = {'Dataset': [f'{Dataset}']*(num_detectors*len_files), 'Number of Keypoints': num_kps
_brisk + num_kps_orb + num_kps_akaze + num_kps_gftt + num_kps_briefstar, 'Detector/Descr
iptor':['BRISK']*len_files  + ['ORB']*len_files  +  ['AKAZE']*len_files +  ['STAR+BRIEF'
]*len_files + ['GFTT+SIFT']*len_files  }
```

In [110]:

```
df = pd.DataFrame.from_dict(d, orient='index')
df = df.transpose()
```

In [111]:

```
df_numkey_15 = df
df_numkey_15['Number of Keypoints'] = df_numkey_15['Number of Keypoints']/(len_files)
```

In [112]:

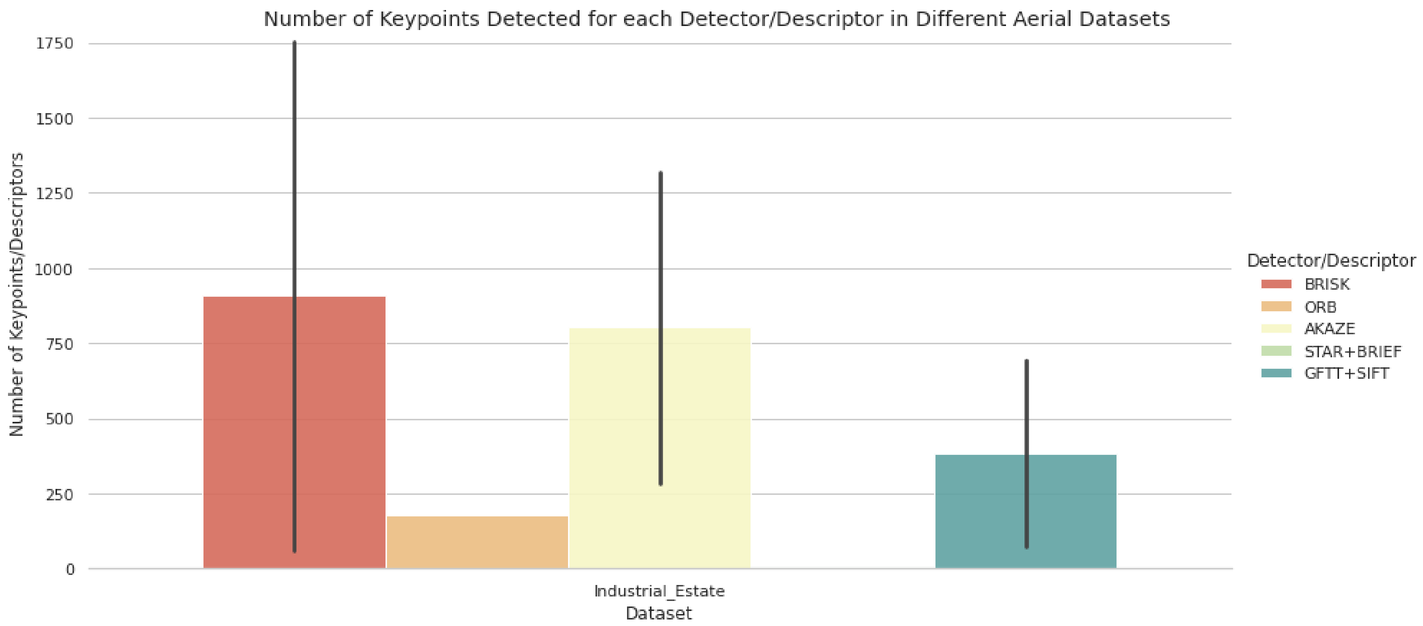
```
import seaborn as sns
sns.set_theme(style='whitegrid')


# Draw a nested barplot by species and sex
g = sns.catplot(
    data=df_numkey_15, kind="bar",
    x="Dataset", y="Number of Keypoints", hue="Detector/Descriptor",
    ci="sd", palette="Spectral", alpha=.9, height=6, aspect=2
)
g.despine(left=True)
```

```
g.set_axis_labels("Dataset", "Number of Keypoints/Descriptors")
g.legend.set_title("Detector/Descriptor")
g.fig.suptitle("Number of Keypoints Detected for each Detector/Descriptor in Different Aerial Datasets")
```

Out[112]:

```
Text(0.5, 0.98, 'Number of Keypoints Detected for each Detector/Descriptor in Different Aerial Datasets')
```



In [113]:

```
df_numkey_15.to_csv(f'drive/MyDrive/Num_Kypoints_15_{Dataset}.csv')
```

In [ ]:

```
#d = {'Dataset': ['University Campus']*(3*len_files), 'Number of Keypoints': num_kps_rootsift + num_kps_superpoint + num_kps_surf, 'Detector/Descriptor':['ROOTSIFT']*101 + ['SuperPoint']*101 + ['SURF']*101  }
#df = pd.DataFrame(data=d)
```

In [ ]:

```
#df_13 = pd.read_csv('drive/MyDrive/Num_Key_13_{Dataset}.csv')
#frames = [df_13, df]
#df_15 = pd.concat(frames)
```

In [ ]:

```
#df_15.to_csv('drive/MyDrive/Num_Key_15_{Dataset}.csv')
```

In [114]:

```
g.savefig(f'drive/MyDrive/Num_Kypoints_15_{Dataset}.png')
```

In [115]:

```
print(len(num_matches_akaze))
```

```
112
```

Didn't get good matches with MSER, so initialize a dummy variable for matches:

In [ ]:

```
num_matches_mser = [0]*len(num_matches_agast)
```

Total Number of Matches Detected for each Detector+Descriptor

In [ ]:

```
#df_match_15['Number of Total Matches'] =  num_matches_agast + num_matches_akaze + num_matches_brisk + num_matches_daisy + num_matches_fast + num_matches_freak + num_matches_gftt + num_matches_kaze + num_matches_mser + num_matches_orb + num_matches_rootsift + num_matches_sift + num_matches_briefstar + num_matches_superpoint+ num_matches_surf+ num_matches_surfsift
d = {'Dataset': [f'{Dataset}']*(num_detectors*(len_files-1)), 'Number of Total Matches': num_matches_agast + num_matches_akaze + num_matches_brisk + num_matches_daisy + num_matches_fast + num_matches_freak + num_matches_gftt + num_matches_kaze + num_matches_mser + num_matches_orb + num_matches_rootsift + num_matches_sift + num_matches_briefstar + num_matches_superpoint+ num_matches_surf, 'Detector/Descriptor':['AGAST+SIFT']*(len_files-1) + ['AKAZE']*(len_files-1) + ['BRISK']*(len_files-1) + ['DAISY+SIFT']*(len_files-1) + ['FAST+SIFT']*(len_files-1) + ['BRISK+FREAK']*(len_files-1) + ['GFTT+SIFT']*(len_files-1) + ['KAZE']*(len_files-1) + ['MSER+SIFT']*(len_files-1) + ['ORB']*(len_files-1) +['RootSIFT']*(len_files-1) +['SIFT']*(len_files-1) + ['STAR+BRIEF']*(len_files-1) +  ['SuperPoint']*(len_files-1) + ['SURF']*(len_files-1) }
df_match_15 = pd.DataFrame(data=d)
df_match_15['Number of Total Matches'] = df_match_15['Number of Total Matches']/(len_files-1)
```

In [ ]:

```
import seaborn as sns
sns.set_theme(style='whitegrid')


# Draw a nested barplot by species and sex
g = sns.catplot(
    data=df_match_15e, kind="bar",
    x="Dataset", y="Number of Total Matches", hue="Detector/Descriptor",
    ci="sd", palette="Spectral", alpha=.9, height=10, aspect=0.5
)
g.despine(left=True)
g.set_axis_labels("Dataset ", "Total Number of Matches b/w Consecutive/Overlapping Images")
g.legend.set_title("Detector/Descriptor")
g.fig.suptitle("Total Number of Matches Detected for each Detector/Descriptor in Different Aerial Datasets")
```

In [ ]:

```
g.savefig(f'drive/MyDrive/Num_Matches_15_{Dataset}.png')
```

In [ ]:

```
#df_match_15.to_csv('drive/MyDrive/Num_Matches_15_{Dataset}.csv')
```

In [ ]:

```
print(min(num_good_matches_agast))
```

Total Number of Good/Robust Matches (NN+Lowe+RANSAC) Detected for each Detector+Descriptor

Didn't get good matches with MSER, so initialize a dummy variable for good matches:

In [ ]:

```
num_good_matches_mser = [0]*len(num_good_matches_agast)
```

In [ ]:

```
df_match_15['Number of Good Matches'] = num_good_matches_agast + num_good_matches_akaze + num_good_matches_brisk + num_good_matches_daisy + num_good_matches_fast + num_good_matches_freak + num_good_matches_gftt + num_good_matches_kaze + num_good_matches_mser + num_good_matches_orb + num_good_matches_rootsift + num_good_matches_sift + num_good_matches
```

```
_briefstar + num_good_matches_superpoint+ num_good_matches_surf
df_match_15['Number of Good Matches'] = df_match_15['Number of Good Matches']/(len_files-
1)
```

In [ ]:

```
import seaborn as sns
sns.set_theme(style='whitegrid')


# Draw a nested barplot by species and sex
g = sns.catplot(
    data=df_match_15, kind="bar",
    x="Dataset", y="Number of Good Matches", hue="Detector/Descriptor",
    ci="sd", palette="Spectral", alpha=.9, height=10, aspect=0.5
)
g.despine(left=True)
g.set_axis_labels("Dataset", "Number of Good Matches b/w Consecutive/Overlapping Images")
g.legend.set_title("Detector/Descriptor")
g.fig.suptitle("Number of Good Matches (Lowe + RANSAC) Detected for each Detector/Descrip
tor in Different Aerial Datasets")
```

In [ ]:

```
g.savefig(f'drive/MyDrive/Num_Good_Matches_15_{Dataset}.png')
```

In [ ]:

```
#df_match_15.to_csv('drive/MyDrive/Num_Good_Matches_15_{Dataset}.csv')
```

Recall Rate for each Detector+Descriptor

In [ ]:

```
df_match_15['Recall Rate of Matches'] = df_match_15['Number of Good Matches']/df_match_15
['Number of Total Matches']
```

In [ ]:

```
import seaborn as sns
sns.set_theme(style='whitegrid')


g = sns.catplot(
    data=df_match_15, kind="bar",
    x="Dataset", y="Recall Rate of Matches", hue="Detector/Descriptor",
    ci="sd", palette="Spectral", alpha=.9, height=10, aspect=0.5
)
g.despine(left=True)
g.set_axis_labels("Dataset", "Precision of Matches")
g.legend.set_title("Detector/Descriptor")
g.fig.suptitle("Recall Rate of Matches Detected (Good/Total) for each Detector/Descriptor
in Different Aerial Datasets (Higher the Better)")
```

In [ ]:

```
g.savefig(f'drive/MyDrive/Recall_Rate_Matches_15_{Dataset}.png')
```

1-Precision Rate for each Detector+Descriptor

In [ ]:

```
df_match_15['1 - Precision Rate of Matches'] = (df_match_15['Number of Total Matches'] -
df_match_15['Number of Good Matches'])/df_match_15['Number of Total Matches']
```

In [ ]:

```
import seaborn as sns
```

```
sns.set_theme(style='whitegrid')


# Draw a nested barplot by species and sex
g = sns.catplot(
    data=df_match_15, kind="bar",
    x="Dataset", y="1 - Precision Rate of Matches", hue="Detector/Descriptor",
    ci="sd", palette="Spectral", alpha=.9, height=10, aspect=0.5
)
g.despine(left=True)
g.set_axis_labels("Dataset (100 Images)", "1 - Precision Rate of Matches")
g.legend.set_title("Detector/Descriptor")
g.fig.suptitle("1 - Precision rate of Matches Detected (False/Total Matches) for each Det
ector/Descriptor in Different Aerial Datasets (Lower the Better)")
```

In [ ]:

```
g.savefig(f'drive/MyDrive/One_minus_Precision_Rate_Matches_15_{Dataset}.png')
```

F-Score for each Detector+Descriptor

In [ ]:

```
df_match_15['F-Score'] = (2* (1 - df_match_15['1 - Precision Rate of Matches']) * df_mat
ch_15['Recall Rate of Matches'])/((1 - df_match_15['1 - Precision Rate of Matches']) + d
f_match_15['Recall Rate of Matches'])
```

In [ ]:

```
import seaborn as sns
sns.set_theme(style='whitegrid')


# Draw a nested barplot by species and sex
g = sns.catplot(
    data=df_match_15, kind="bar",
    x="Dataset", y="F-Score", hue="Detector/Descriptor",
    ci="sd", palette="Spectral", alpha=.9, height=10, aspect=0.5
)
g.despine(left=True)
g.set_axis_labels("Dataset", "F-Score")
g.legend.set_title("Detector/Descriptor")
g.fig.suptitle("F-Score of Matches Detected (2*P*R/P+R) for each Detector/Descriptor in D
ifferent Aerial Datasets (Higher the Better)")
```

In [ ]:

```
g.savefig(f'drive/MyDrive/F_Score_Rate_Matches_15_{Dataset}.png')
```

In [ ]:

```
df_match_15.to_csv(f'drive/MyDrive/All_metrics_15_{Dataset}.csv')
```

Time for each Detector+Descriptor

In [ ]:

```
d = {'Dataset': [f'{Dataset}']*(num_detectors), 'Time':  [time_all[7]] + [time_all[3]] +
[time_all[0]] + [time_all[5]] + [time_all[10]] + [time_all[8]] + [time_all[9]] + [time_a
ll[2]] + [time_all[6]] + [time_all[1]] + [time_all[13]] + [time_all[11]] + [time_all[4]]
+ [time_all[14]] + [time_all[12]] , 'Detector/Descriptor':['AGAST+SIFT'] + ['AKAZE'] + [
'BRISK']*(1) + ['DAISY+SIFT']*(1) + ['FAST+SIFT']*(1) + ['BRISK+FREAK']*(1) + ['GFTT+SIF
T']*(1) + ['KAZE']*(1) + ['MSER+SIFT']*(1) + ['ORB']*(1) +['RootSIFT']*(1) +['SIFT']*(1
) + ['STAR+BRIEF']*(1) +  ['SuperPoint']*(1) + ['SURF']*(1)}
df_time_15 = pd.DataFrame(data=d)
```

In [ ]:

```
import seaborn as sns
sns.set_theme(style='whitegrid')


# Draw a nested barplot by species and sex
g = sns.catplot(
    data=df_time_15, kind="bar",
    x="Dataset", y="Time", hue="Detector/Descriptor",
    ci="sd", palette="Spectral", alpha=.9, height=10, aspect=0.5
)
g.despine(left=True)
g.set_axis_labels("Dataset", "Time (in sec)")
g.legend.set_title("Detector/Descriptor")
g.fig.suptitle("Time taken during Feature Extraction by each Detector/Descriptor in Different Aerial Datasets (Lower the Better)")
```

In [ ]:

```
g.savefig(f'drive/MyDrive/Time_15_{Dataset}.png')
```

In [ ]:

```
df_time_15.to_csv(f'drive/MyDrive/Time_15_{Dataset}.csv')
```

In [ ]:

```

```