In [3]:

```
!pip install torchsummary
```

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

In [4]:

```
import numpy as np

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

In [5]:

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

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

```
        ratio = minval1/minval2

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

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

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

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

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

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

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

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

In [1]:

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

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

In [2]:

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

In [6]:

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

folder_path = '../input/uni-campus-dataset/RGB-img/img/'
left_files_path_rev = []
right_files_path = []
for file in files_all[:121]:
    left_files_path_rev.append(folder_path + file)


left_files_path = left_files_path_rev[::-1]


for file in files_all[121:200]:
    right_files_path.append(folder_path + file)
```

In [8]:

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


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


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

```
100%|██████████| 121/121 [01:31<00:00,  1.33it/s]
100%|██████████| 79/79 [01:02<00:00,  1.27it/s]
```

In [9]:

```
images_left_bgr_no_enhance = []
images_right_bgr_no_enhance = []

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

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

```
100%|██████████| 121/121 [00:50<00:00,  2.40it/s]
100%|██████████| 79/79 [00:32<00:00,  2.41it/s]
```

In [ ]:

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

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


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


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

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

In [ ]:

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

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

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

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

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

In [10]:

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

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

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

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

```
100%|██████████| 121/121 [14:36<00:00,  7.24s/it]
100%|██████████| 79/79 [09:44<00:00,  7.40s/it]
```

In [10]:

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

In [ ]:

```
akaze = cv2.AKAZE_create()
keypoints_all_left_akaze = []
descriptors_all_left_akaze = []
points_all_left_akaze=[]

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

for imgs in tqdm(images_left_bgr):
    kpt = akaze.detect(imgs,None)
    kpt,descrip = akaze.compute(imgs, kpt)
    keypoints_all_left_akaze.append(kpt)
    descriptors_all_left_akaze.append(descrip)
    points_all_left_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
for imgs in tqdm(images_right_bgr):
    kpt = akaze.detect(imgs,None)
    kpt,descrip = akaze.compute(imgs, kpt)
    keypoints_all_right_akaze.append(kpt)
    descriptors_all_right_akaze.append(descrip)
    points_all_right_akaze.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

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


keypoints_all_right_star = []
```

```
descriptors_all_right_brief = []
points_all_right_star=[]

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

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

In [ ]:

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


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

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

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

In [ ]:

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

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


for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = mser.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_mser.append(kpt)
    descriptors_all_right_mser.append(descrip)
```

```
    points_all_right_mser.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

```
agast = cv2.AgastFeatureDetector_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_agast = []
descriptors_all_left_agast = []
points_all_left_agast=[]

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


for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = agast.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_agast.append(kpt)
    descriptors_all_left_agast.append(descrip)
    points_all_left_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))


for imgs in tqdm(images_right_bgr_no_enhance):
    kpt = agast.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_agast.append(kpt)
    descriptors_all_right_agast.append(descrip)
    points_all_right_agast.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [ ]:

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

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

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

In [ ]:

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

keypoints_all_right_gftt = []
descriptors_all_right_gftt = []
points_all_right_gftt=[]
for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = gftt.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_gftt.append(kpt)
    descriptors_all_left_gftt.append(descrip)
```

```
    points_all_left_gftt.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

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

In [ ]:

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

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

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

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

In [11]:

```
surf = cv2.xfeatures2d.SURF_create()
sift = cv2.xfeatures2d.SIFT_create()
keypoints_all_left_surfsift = []
descriptors_all_left_surfsift = []
points_all_left_surfsift=[]

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

for imgs in tqdm(images_left_bgr_no_enhance):
    kpt = surf.detect(imgs,None)
    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_left_surfsift.append(kpt)
    descriptors_all_left_surfsift.append(descrip)
    points_all_left_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))

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

    kpt,descrip = sift.compute(imgs, kpt)
    keypoints_all_right_surfsift.append(kpt)
    descriptors_all_right_surfsift.append(descrip)
    points_all_right_surfsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

```
100%|██████████| 121/121 [19:52<00:00,  9.85s/it]
100%|██████████| 79/79 [13:46<00:00, 10.46s/it]
```

In [10]:

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

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

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

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

```
100%|██████████| 121/121 [02:47<00:00,  1.39s/it]
100%|██████████| 79/79 [01:52<00:00,  1.42s/it]
```

In [ ]:

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

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

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

In [ ]:

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

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

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

In [ ]:

```
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 [ ]:

```
sift = cv2.xfeatures2d.SIFT_create()
rootsift = RootSIFT()
keypoints_all_left_rootsift = []
descriptors_all_left_rootsift = []
points_all_left_rootsift=[]

keypoints_all_right_rootsift = []
descriptors_all_right_rootsift = []
points_all_right_rootsift=[]

for imgs in tqdm(images_left_bgr):
    kpt = sift.detect(imgs,None)
    kpt,descrip = rootsift.compute(imgs, kpt)
    keypoints_all_left_rootsift.append(kpt)
    descriptors_all_left_rootsift.append(descrip)
    points_all_left_rootsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
for imgs in tqdm(images_right_bgr):
    kpt = sift.detect(imgs,None)
    kpt,descrip = rootsift.compute(imgs, kpt)
    keypoints_all_right_rootsift.append(kpt)
    descriptors_all_right_rootsift.append(descrip)
    points_all_right_rootsift.append(np.asarray([[p.pt[0], p.pt[1]] for p in kpt]))
```

In [12]:

```
!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 [13]:

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

In [14]:

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

In [15]:

```
torch.cuda.empty_cache()
class SuperPointNet(nn.Module):
    def __init__(self):
        super(SuperPointNet,self).__init__()
        self.relu = nn.ReLU(inplace=True)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)
        c1,c2,c3,c4,c5,d1 = 64,64,128,128,256,256
        self.conv1a = nn.Conv2d(1,c1,kernel_size=3,stride=1,padding=1)
        self.conv1b = nn.Conv2d(c1,c1,kernel_size=3,stride=1,padding=1)
```

```
        self.conv2a = nn.Conv2d(c1,c2,kernel_size=3,stride=1,padding=1)
        self.conv2b = nn.Conv2d(c2,c2,kernel_size=3,stride=1,padding=1)
        self.conv3a = nn.Conv2d(c2,c3,kernel_size=3,stride=1,padding=1)
        self.conv3b = nn.Conv2d(c3,c3,kernel_size=3,stride=1,padding=1)
        self.conv4a = nn.Conv2d(c3,c4,kernel_size=3,stride=1,padding=1)
        self.conv4b = nn.Conv2d(c4,c4,kernel_size=3,stride=1,padding=1)
        self.convPa = nn.Conv2d(c4,c5,kernel_size=3,stride=1,padding=1)
        self.convPb = nn.Conv2d(c5,65,kernel_size=1,stride=1,padding=0)
        self.convDa = nn.Conv2d(c4,c5,kernel_size=3,stride=1,padding=1)


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

    def forward(self,x):
        x = self.relu(self.conv1a(x))
        x = self.relu(self.conv1b(x))
        x = self.pool(x)
        x = self.relu(self.conv2a(x))
        x = self.relu(self.conv2b(x))
        x = self.pool(x)
        x  = self.relu(self.conv3a(x))
        x = self.relu(self.conv3b(x))
        x = self.pool(x)
        x = self.relu(self.conv4a(x))
        x = self.relu(self.conv4b(x))
        cPa = self.relu(self.convPa(x))
        semi = self.convPb(cPa)
        cDa = self.relu(self.convDa(x))
        desc = self.convDb(cDa)
        dn = torch.norm(desc,p=2,dim=1)
        desc = desc.div(torch.unsqueeze(dn,1))
        return semi,desc


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

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

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

```
        if grid[pt[1], pt[0]] == 1:
            grid[pt[1]-pad:pt[1]+pad+1, pt[0]-pad:pt[0]+pad+1]=0


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

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

def run(self,img):
    assert img.ndim == 2
    assert img.dtype == np.float32
    H,W = img.shape[0], img.shape[1]
    inp = img.copy()
    inp = (inp.reshape(1,H,W))
    inp = torch.from_numpy(inp)
    inp = torch.autograd.Variable(inp).view(1,1,H,W)
    if self.cuda:
        inp = inp.cuda()
    outs = self.net.forward(inp)
    semi,coarse_desc = outs[0],outs[1]
    semi = semi.data.cpu().numpy().squeeze()


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

    return heatmap,coarse_desc

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

    pts = np.zeros((3,len(xs)))
    pts[0,:] = ys
    pts[1,:] = xs
    pts[2,:] = heat_map[xs,ys]
    pts,_ = self.nms_fast(pts,H,W,dist_thresh=self.nms_dist)
    inds = np.argsort(pts[2,:])
    pts = pts[:,inds[::-1]]
    bord = self.border_remove
    toremoveW = np.logical_or(pts[0,:] < bord, pts[0,:] >= (W-bord))
    toremoveH = np.logical_or(pts[1,:] < bord, pts[0,:] >= (H-bord))
    toremove = np.logical_or(toremoveW, toremoveH)
    pts = pts[:,~toremove]
    pts = pts[:,0:sampled]
    D = coarse_desc.shape[1]
    if pts.shape[1] == 0:
        desc = np.zeros((D,0))
    else:
        samp_pts = torch.from_numpy(pts[:2,:].copy())
        samp_pts[0,:] = (samp_pts[0,:] / (float(W)/2.))-1.
        samp_pts[1,:] = (samp_pts[1,:] / (float(W)/2.))-1.
```

```
            samp_pts = samp_pts.transpose(0,1).contiguous()
            samp_pts = samp_pts.view(1,1,-1,2)
            samp_pts = samp_pts.float()
            if self.cuda:
                samp_pts = samp_pts.cuda()
            desc = nn.functional.grid_sample(coarse_desc, samp_pts)
            desc = desc.data.cpu().numpy().reshape(D,-1)
            desc /= np.linalg.norm(desc,axis=0)[np.newaxis,:]
        return pts,desc
```

In [16]:

```
print('Load pre trained network')
fe = SuperPointFrontend(weights_path = weights_path, nms_dist = 4, conf_thresh  = 0.015,
nn_thresh=0.7,
                        cuda = False)
print('Successfully  loaded pretrained network')
```

```
Load pre trained network
Successfully  loaded pretrained network
```

In [ ]:

```
keypoint_all_left_superpoint = []
descriptor_all_left_superpoint = []
point_all_left_superpoint = []

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

for ifpth in tqdm(images_left):
    heatmap1, coarse_desc1 = fe.run(ifpth)
    pts_1, desc_1 = fe.key_pt_sampling(ifpth,heatmap1,coarse_desc1,2000)

    keypoint_all_left_superpoint.append(to_kpts(pts_1.T))
    descriptor_all_left_superpoint.append(desc_1.T)
    point_all_left_superpoint.append(pts_1.T)





for rfpth in tqdm(images_right):
    heatmap1, coarse_desc1 = fe.run(rfpth)
    pts_1, desc_1 = fe.key_pt_sampling(rfpth,heatmap1,coarse_desc1,2000)

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [16]:

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

```
100%|██████████| 200/200 [00:00<00:00, 477167.69it/s]
```

In [ ]:

```
num_kps_akaze  = []




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

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [18]:

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

```
100%|██████████| 200/200 [00:00<00:00, 449791.31it/s]
```

In [ ]:

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

In [17]:

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

```
100%|██████████| 200/200 [00:00<00:00, 301531.56it/s]
```

In [ ]:

```
num_kps_agast  = []
```

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

In [18]:

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

In [19]:

```
def get_Hmatrix(imgs,keypts,pts,descripts,ratio=0.8,thresh=4,disp=False):
    FLANN_INDEX_KDTREE = 2
    index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
    search_params = dict(checks=50)
    flann = cv2.FlannBasedMatcher(index_params, search_params)
    #flann = cv2.BFMatcher()
    lff1 = np.float32(descripts[0])
    lff = np.float32(descripts[1])
    matches_lf1_lf = flann.knnMatch(lff1, lff, k=2)
    print("\nNumber of matches",len(matches_lf1_lf))
    matches_4 = []
    ratio = ratio
    # loop over the raw matches
    for m in matches_lf1_lf:
        # ensure the distance is within a certain ratio of each
        # other (i.e. Lowe’s ratio test)
        if len(m) == 2 and m[0].distance < m[1].distance * ratio:

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

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

    inlier_matchset = np.array(matches_4)[inliers.astype(bool)].tolist()
    print("Number of Robust matches",len(inlier_matchset))
    print("\n")
    '''
    if len(inlier_matchset)<50:
        matches_4 = []
        ratio = 0.67
        # loop over the raw matches
        for m in matches_lf1_lf:
           # ensure the distance is within a certain ratio of each
```

```
            # other (i.e. Lowe’s ratio test)
            if len(m) == 2 and m[0].distance < m[1].distance * ratio:
            #matches_1.append((m[0].trainIdx, m[0].queryIdx))
            matches_4.append(m[0])
        print("Number of matches After Lowe's Ratio New",len(matches_4))
        matches_idx = np.array([m.queryIdx for m in matches_4])
        imm1_pts = np.array([keypts[0][idx].pt for idx in matches_idx])
        matches_idx = np.array([m.trainIdx for m in matches_4])
        imm2_pts = np.array([keypts[1][idx].pt for idx in matches_idx])
        Hn,inliers = compute_homography_fast_other(imm1_pts,imm2_pts)
        inlier_matchset = np.array(matches_4)[inliers.astype(bool)].tolist()
        print("Number of Robust matches New",len(inlier_matchset))
        print("\n")
       '''
    #H=compute_Homography(imm1_pts,imm2_pts)
    #Robustly estimate Homography 1 using RANSAC
    #Hn=RANSAC_alg(keypts[0] ,keypts[1], matches_4, nRANSAC=1500, RANSACthresh=6)
    #global inlier_matchset
    if disp==True:
        dispimg1=cv2.drawMatches(imgs[0], keypts[0], imgs[1], keypts[1], inlier_matchset
, None,flags=2)
        displayplot(dispimg1,'Robust Matching between Reference Image and Right Image ')
    return Hn/Hn[2,2], len(matches_lf1_lf), len(inlier_matchset)
```

In [20]:

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

In [ ]:

```
H_left_brisk = []
H_right_brisk = []

num_matches_brisk = []
num_good_matches_brisk = []

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

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_brisk[j:j+2][::-1],points_all_left_brisk[j:j+2][::-1],descriptors_all_left_brisk[j:j+2]
[::-1])
    H_left_brisk.append(H_a)
    num_matches_brisk.append(matches)
    num_good_matches_brisk.append(gd_matches)

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

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

In [ ]:

```
H_left_orb = []
H_right_orb = []

num_matches_orb = []
num_good_matches_orb = []

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

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

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

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

In [ ]:

```
H_left_akaze = []
H_right_akaze = []

num_matches_akaze = []
num_good_matches_akaze = []

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

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

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

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

In [20]:

```
H_left_kaze = []
H_right_kaze = []

num_matches_kaze = []
num_good_matches_kaze = []

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

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

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

```
    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_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)
```

```
  1%|          | 1/121 [00:01<02:05,  1.05s/it]

Number of matches 12007
Number of matches After Lowe's Ratio 3694
Number of Robust matches 2247


  2%|          | 2/121 [00:02<02:04,  1.04s/it]

Number of matches 12841
Number of matches After Lowe's Ratio 3379
Number of Robust matches 2291


  2%|          | 3/121 [00:03<01:56,  1.01it/s]

Number of matches 15771
Number of matches After Lowe's Ratio 3981
Number of Robust matches 2640


  3%|          | 4/121 [00:04<02:04,  1.06s/it]

Number of matches 19597
Number of matches After Lowe's Ratio 4596
Number of Robust matches 3192


  4%|          | 5/121 [00:05<02:22,  1.23s/it]

Number of matches 22711
Number of matches After Lowe's Ratio 3552
Number of Robust matches 1857


  5%|          | 6/121 [00:07<02:50,  1.48s/it]

Number of matches 22615
Number of matches After Lowe's Ratio 5781
Number of Robust matches 2800


  6%|          | 7/121 [00:09<03:01,  1.59s/it]

Number of matches 25385
Number of matches After Lowe's Ratio 3678
Number of Robust matches 1910


  7%|          | 8/121 [00:11<03:13,  1.72s/it]

Number of matches 23521
Number of matches After Lowe's Ratio 1437
Number of Robust matches 617


  7%|          | 9/121 [00:13<03:29,  1.87s/it]

Number of matches 21785
Number of matches After Lowe's Ratio 4419
```

```
Number of matches After Lowe's Ratio 1119
Number of Robust matches 2399


  8%|          | 10/121 [00:15<03:20,  1.81s/it]

Number of matches 18996
Number of matches After Lowe's Ratio 5151
Number of Robust matches 2885


  9%|          | 11/121 [00:16<03:06,  1.70s/it]

Number of matches 20037
Number of matches After Lowe's Ratio 4324
Number of Robust matches 2426


 10%|          | 12/121 [00:18<02:58,  1.64s/it]

Number of matches 20009
Number of matches After Lowe's Ratio 5310
Number of Robust matches 2957


 11%|          | 13/121 [00:19<02:51,  1.59s/it]

Number of matches 17658
Number of matches After Lowe's Ratio 5422
Number of Robust matches 3530


 12%|          | 14/121 [00:21<02:42,  1.52s/it]

Number of matches 12178
Number of matches After Lowe's Ratio 2690
Number of Robust matches 1821


 12%|          | 15/121 [00:21<02:18,  1.31s/it]

Number of matches 11226
Number of matches After Lowe's Ratio 3517
Number of Robust matches 2137


 13%|          | 16/121 [00:22<02:00,  1.15s/it]

Number of matches 11003
Number of matches After Lowe's Ratio 1597
Number of Robust matches 849


 14%|          | 17/121 [00:23<01:47,  1.03s/it]

Number of matches 13601
Number of matches After Lowe's Ratio 3014
Number of Robust matches 1571


 15%|          | 18/121 [00:24<01:53,  1.10s/it]

Number of matches 20716
Number of matches After Lowe's Ratio 2648
Number of Robust matches 1614
```

```
 16%|▉         | 19/121 [00:26<02:07,  1.25s/it]

Number of matches 20830
Number of matches After Lowe's Ratio 7656
Number of Robust matches 4948


 17%|▉         | 20/121 [00:28<02:35,  1.54s/it]

Number of matches 19609
Number of matches After Lowe's Ratio 4537
Number of Robust matches 2729


 17%|▉         | 21/121 [00:30<02:33,  1.53s/it]

Number of matches 19798
Number of matches After Lowe's Ratio 6820
Number of Robust matches 4529


 18%|▉         | 22/121 [00:31<02:31,  1.53s/it]

Number of matches 20129
Number of matches After Lowe's Ratio 4566
Number of Robust matches 3509


 19%|▉         | 23/121 [00:33<02:29,  1.53s/it]

Number of matches 20334
Number of matches After Lowe's Ratio 4466
Number of Robust matches 2905


Number of matches 20079
Number of matches After Lowe's Ratio 3759
 20%|▉         | 24/121 [00:34<02:36,  1.61s/it]

Number of Robust matches 2782


 21%|▉         | 25/121 [00:36<02:33,  1.60s/it]

Number of matches 21359
Number of matches After Lowe's Ratio 4441
Number of Robust matches 2613


 21%|▉         | 26/121 [00:38<02:33,  1.62s/it]

Number of matches 22554
Number of matches After Lowe's Ratio 5431
Number of Robust matches 2713


 22%|▉         | 27/121 [00:39<02:35,  1.66s/it]

Number of matches 23245
Number of matches After Lowe's Ratio 6967
Number of Robust matches 3046


 23%|▉         | 28/121 [00:41<02:42,  1.75s/it]
```

```
Number of matches 22020
Number of matches After Lowe's Ratio 4038
Number of Robust matches 1673
```

```
 24%|[redacted]        | 29/121 [00:43<02:39,  1.74s/it]
```

```
Number of matches 21755
Number of matches After Lowe's Ratio 7620
Number of Robust matches 2960
```

```
 25%|[redacted]        | 30/121 [00:45<02:36,  1.71s/it]
```

```
Number of matches 22293
Number of matches After Lowe's Ratio 4581
Number of Robust matches 1644
```

```
 26%|[redacted]        | 31/121 [00:47<02:40,  1.78s/it]
```

```
Number of matches 20480
Number of matches After Lowe's Ratio 4495
Number of Robust matches 1740
```

```
 26%|[redacted]        | 32/121 [00:48<02:37,  1.77s/it]
```

```
Number of matches 21219
Number of matches After Lowe's Ratio 4377
Number of Robust matches 1703
```

```
 27%|[redacted]        | 33/121 [00:50<02:33,  1.74s/it]
```

```
Number of matches 23237
Number of matches After Lowe's Ratio 4893
Number of Robust matches 2047
```

```
 28%|[redacted]        | 34/121 [00:52<02:35,  1.79s/it]
```

```
Number of matches 25532
Number of matches After Lowe's Ratio 5603
Number of Robust matches 2268
```

```
 29%|[redacted]        | 35/121 [00:54<02:47,  1.94s/it]
```

```
Number of matches 31041
Number of matches After Lowe's Ratio 3372
Number of Robust matches 1228
```

```
 30%|[redacted]        | 36/121 [00:57<03:13,  2.28s/it]
```

```
Number of matches 26455
Number of matches After Lowe's Ratio 424
Number of Robust matches 62
```

```
 31%|[redacted]        | 37/121 [01:00<03:18,  2.36s/it]
```

```
Number of matches 28555
Number of matches After Lowe's Ratio 4172
Number of Robust matches 1761
```

```
 31%|[redacted]       | 38/121 [01:02<03:15,  2.36s/it]

Number of matches 20882
Number of matches After Lowe's Ratio 1931
Number of Robust matches 691


 32%|[redacted]       | 39/121 [01:04<02:58,  2.18s/it]

Number of matches 20959
Number of matches After Lowe's Ratio 4438
Number of Robust matches 2249


 33%|[redacted]       | 40/121 [01:06<02:43,  2.02s/it]

Number of matches 21771
Number of matches After Lowe's Ratio 5486
Number of Robust matches 2175


 34%|[redacted]       | 41/121 [01:08<02:38,  1.98s/it]

Number of matches 22352
Number of matches After Lowe's Ratio 5761
Number of Robust matches 2533


 35%|[redacted]       | 42/121 [01:09<02:34,  1.95s/it]

Number of matches 22841
Number of matches After Lowe's Ratio 6950
Number of Robust matches 2911


 36%|[redacted]       | 43/121 [01:11<02:29,  1.91s/it]

Number of matches 24425
Number of matches After Lowe's Ratio 6246
Number of Robust matches 2370


 36%|[redacted]       | 44/121 [01:13<02:29,  1.94s/it]

Number of matches 25661
Number of matches After Lowe's Ratio 7562
Number of Robust matches 3071


 37%|[redacted]       | 45/121 [01:15<02:30,  1.98s/it]

Number of matches 25368
Number of matches After Lowe's Ratio 7188
Number of Robust matches 3404


 38%|[redacted]       | 46/121 [01:18<02:33,  2.04s/it]

Number of matches 23731
Number of matches After Lowe's Ratio 6846
Number of Robust matches 3636


 39%|[redacted]       | 47/121 [01:20<02:31,  2.05s/it]
```

```
Number of matches 22744
Number of matches After Lowe's Ratio 6184
Number of Robust matches 4132
```

```
 40%|[redacted]      | 48/121 [01:21<02:23,  1.96s/it]
```

```
Number of matches 22557
Number of matches After Lowe's Ratio 6807
Number of Robust matches 4496
```

```
 40%|[redacted]      | 49/121 [01:23<02:19,  1.93s/it]
```

```
Number of matches 21002
Number of matches After Lowe's Ratio 5313
Number of Robust matches 3909
```

```
 41%|[redacted]      | 50/121 [01:25<02:09,  1.83s/it]
```

```
Number of matches 20707
Number of matches After Lowe's Ratio 6708
Number of Robust matches 5350
```

```
 42%|[redacted]      | 51/121 [01:26<02:03,  1.76s/it]
```

```
Number of matches 22231
Number of matches After Lowe's Ratio 6755
Number of Robust matches 4063
```

```
 43%|[redacted]      | 52/121 [01:28<02:04,  1.81s/it]
```

```
Number of matches 23306
Number of matches After Lowe's Ratio 8614
Number of Robust matches 6256
```

```
 44%|[redacted]      | 53/121 [01:30<02:08,  1.89s/it]
```

```
Number of matches 23113
Number of matches After Lowe's Ratio 8091
Number of Robust matches 5404
```

```
 45%|[redacted]      | 54/121 [01:32<02:02,  1.83s/it]
```

```
Number of matches 18219
Number of matches After Lowe's Ratio 3360
Number of Robust matches 2188
```

```
 45%|[redacted]      | 55/121 [01:34<02:01,  1.84s/it]
```

```
Number of matches 21377
Number of matches After Lowe's Ratio 4503
Number of Robust matches 3160
```

```
 46%|[redacted]      | 56/121 [01:36<01:57,  1.81s/it]
```

```
Number of matches 17969
Number of matches After Lowe's Ratio 1538
Number of Robust matches 779
```

```
 47%|▌▌▌▌▌      | 57/121 [01:37<01:46,  1.66s/it]

Number of matches 19990
Number of matches After Lowe's Ratio 2938
Number of Robust matches 1970


 48%|▌▌▌▌▌      | 58/121 [01:39<01:43,  1.64s/it]

Number of matches 22253
Number of matches After Lowe's Ratio 5743
Number of Robust matches 4018


 49%|▌▌▌▌▌      | 59/121 [01:40<01:45,  1.70s/it]

Number of matches 19340
Number of matches After Lowe's Ratio 5123
Number of Robust matches 3768


 50%|▌▌▌▌▌      | 60/121 [01:42<01:39,  1.64s/it]

Number of matches 17684
Number of matches After Lowe's Ratio 3021
Number of Robust matches 1592


 50%|▌▌▌▌▌      | 61/121 [01:43<01:35,  1.58s/it]

Number of matches 17737
Number of matches After Lowe's Ratio 2165
Number of Robust matches 946


 51%|▌▌▌▌▌      | 62/121 [01:45<01:29,  1.52s/it]

Number of matches 23066
Number of matches After Lowe's Ratio 1995
Number of Robust matches 884


 52%|▌▌▌▌▌      | 63/121 [01:47<01:32,  1.59s/it]

Number of matches 20398
Number of matches After Lowe's Ratio 666
Number of Robust matches 152


 53%|▌▌▌▌▌      | 64/121 [01:48<01:30,  1.58s/it]

Number of matches 19469
Number of matches After Lowe's Ratio 5080
Number of Robust matches 2593


 54%|▌▌▌▌▌      | 65/121 [01:50<01:30,  1.61s/it]

Number of matches 20021
Number of matches After Lowe's Ratio 5451
Number of Robust matches 2983


 55%|▌▌▌▌▌      | 66/121 [01:51<01:29,  1.64s/it]
```

```
Number of matches 20426
Number of matches After Lowe's Ratio 5269
Number of Robust matches 2659
```

```
 55%|[redacted]     | 67/121 [01:53<01:28,  1.64s/it]
```

```
Number of matches 22778
Number of matches After Lowe's Ratio 5856
Number of Robust matches 3636
```

```
 56%|[redacted]     | 68/121 [01:55<01:27,  1.66s/it]
```

```
Number of matches 17408
Number of matches After Lowe's Ratio 3030
Number of Robust matches 1851
```

```
 57%|[redacted]     | 69/121 [01:56<01:25,  1.65s/it]
```

```
Number of matches 23670
Number of matches After Lowe's Ratio 4671
Number of Robust matches 3220
```

```
 58%|[redacted]     | 70/121 [01:58<01:26,  1.69s/it]
```

```
Number of matches 18203
Number of matches After Lowe's Ratio 2667
Number of Robust matches 1474
```

```
 59%|[redacted]     | 71/121 [02:00<01:20,  1.61s/it]
```

```
Number of matches 22285
Number of matches After Lowe's Ratio 5441
Number of Robust matches 3760
```

```
 60%|[redacted]     | 72/121 [02:01<01:21,  1.66s/it]
```

```
Number of matches 22018
Number of matches After Lowe's Ratio 6586
Number of Robust matches 4498
```

```
 60%|[redacted]     | 73/121 [02:04<01:26,  1.80s/it]
```

```
Number of matches 24330
Number of matches After Lowe's Ratio 6329
Number of Robust matches 4347
```

```
 61%|[redacted]     | 74/121 [02:06<01:30,  1.92s/it]
```

```
Number of matches 24904
Number of matches After Lowe's Ratio 9134
Number of Robust matches 6785
```

```
 62%|[redacted]     | 75/121 [02:08<01:32,  2.01s/it]
```

```
Number of matches 23410
Number of matches After Lowe's Ratio 6908
Number of Robust matches 5172
```

```
 63%|▉▉▉▉▉▉▉    | 76/121 [02:10<01:29,  2.00s/it]

Number of matches 21871
Number of matches After Lowe's Ratio 7111
Number of Robust matches 5506


 64%|▉▉▉▉▉▉▉    | 77/121 [02:12<01:23,  1.91s/it]

Number of matches 21387
Number of matches After Lowe's Ratio 6427
Number of Robust matches 4502


 64%|▉▉▉▉▉▉▉    | 78/121 [02:13<01:20,  1.86s/it]

Number of matches 21095
Number of matches After Lowe's Ratio 7139
Number of Robust matches 5723


Number of matches 21423
Number of matches After Lowe's Ratio 8093
 65%|▉▉▉▉▉▉▉    | 79/121 [02:15<01:18,  1.86s/it]

Number of Robust matches 6295


 66%|▉▉▉▉▉▉▉    | 80/121 [02:17<01:13,  1.79s/it]

Number of matches 20665
Number of matches After Lowe's Ratio 6626
Number of Robust matches 4991


 67%|▉▉▉▉▉▉▉    | 81/121 [02:18<01:09,  1.74s/it]

Number of matches 21776
Number of matches After Lowe's Ratio 5960
Number of Robust matches 4099


 68%|▉▉▉▉▉▉▉    | 82/121 [02:20<01:07,  1.72s/it]

Number of matches 21075
Number of matches After Lowe's Ratio 5775
Number of Robust matches 3845


 69%|▉▉▉▉▉▉▉    | 83/121 [02:22<01:06,  1.74s/it]

Number of matches 22361
Number of matches After Lowe's Ratio 5762
Number of Robust matches 3162


 69%|▉▉▉▉▉▉▉    | 84/121 [02:24<01:04,  1.74s/it]

Number of matches 21739
Number of matches After Lowe's Ratio 4962
Number of Robust matches 3311
```

```
 70%|▉▉▉▉▉▉▉   | 85/121 [02:26<01:05,  1.82s/it]

Number of matches 24424
Number of matches After Lowe's Ratio 4510
Number of Robust matches 2714


 71%|▉▉▉▉▉▉▉   | 86/121 [02:28<01:04,  1.85s/it]

Number of matches 21469
Number of matches After Lowe's Ratio 4727
Number of Robust matches 2542


 72%|▉▉▉▉▉▉▉   | 87/121 [02:30<01:03,  1.86s/it]

Number of matches 22552
Number of matches After Lowe's Ratio 4868
Number of Robust matches 2498


 73%|▉▉▉▉▉▉▉   | 88/121 [02:31<01:00,  1.83s/it]

Number of matches 21879
Number of matches After Lowe's Ratio 4758
Number of Robust matches 2208


 74%|▉▉▉▉▉▉▉   | 89/121 [02:33<00:57,  1.80s/it]

Number of matches 25013
Number of matches After Lowe's Ratio 3136
Number of Robust matches 1286


 74%|▉▉▉▉▉▉▉   | 90/121 [02:35<01:00,  1.94s/it]

Number of matches 24466
Number of matches After Lowe's Ratio 4386
Number of Robust matches 2112


 75%|▉▉▉▉▉▉▉   | 91/121 [02:37<01:00,  2.00s/it]

Number of matches 24798
Number of matches After Lowe's Ratio 2439
Number of Robust matches 1039


 76%|▉▉▉▉▉▉▉   | 92/121 [02:40<01:01,  2.14s/it]

Number of matches 21670
Number of matches After Lowe's Ratio 1128
Number of Robust matches 320


 77%|▉▉▉▉▉▉▉   | 93/121 [02:42<00:56,  2.01s/it]

Number of matches 21083
Number of matches After Lowe's Ratio 4219
Number of Robust matches 2086


 78%|▉▉▉▉▉▉▉   | 94/121 [02:43<00:52,  1.94s/it]

Number of matches 19041
Number of matches After Lowe's Ratio 4627
```

```
Number of Robust matches 2454

 79%|▉▉▉▉▉▉▉▉  | 95/121 [02:45<00:46,  1.80s/it]

Number of matches 21135
Number of matches After Lowe's Ratio 4252
Number of Robust matches 2256

 79%|▉▉▉▉▉▉▉▉  | 96/121 [02:47<00:46,  1.86s/it]

Number of matches 25378
Number of matches After Lowe's Ratio 5534
Number of Robust matches 2644

 80%|▉▉▉▉▉▉▉▉  | 97/121 [02:49<00:47,  1.96s/it]

Number of matches 28928
Number of matches After Lowe's Ratio 5133
Number of Robust matches 2147

 81%|▉▉▉▉▉▉▉▉  | 98/121 [02:52<00:50,  2.18s/it]

Number of matches 31135
Number of matches After Lowe's Ratio 7071
Number of Robust matches 3025

 82%|▉▉▉▉▉▉▉▉  | 99/121 [02:54<00:50,  2.30s/it]

Number of matches 28082
Number of matches After Lowe's Ratio 6764
Number of Robust matches 2843


Number of matches 24578
Number of matches After Lowe's Ratio 6492
 83%|▉▉▉▉▉▉▉▉  | 101/121 [02:59<00:46,  2.32s/it]

Number of Robust matches 3767


Number of matches 22760
Number of matches After Lowe's Ratio 6545
Number of Robust matches 3920

 84%|▉▉▉▉▉▉▉▉  | 102/121 [03:01<00:40,  2.15s/it]

Number of matches 21201
Number of matches After Lowe's Ratio 6238
Number of Robust matches 4487

 85%|▉▉▉▉▉▉▉▉  | 103/121 [03:02<00:36,  2.00s/it]

Number of matches 21619
Number of matches After Lowe's Ratio 7159
Number of Robust matches 5122
```

```
 86%|██████████ | 104/121 [03:04<00:33,  1.94s/it]

Number of matches 23323
Number of matches After Lowe's Ratio 6293
Number of Robust matches 4108


 87%|██████████ | 105/121 [03:06<00:30,  1.91s/it]

Number of matches 24274
Number of matches After Lowe's Ratio 7448
Number of Robust matches 4029


 88%|██████████ | 106/121 [03:08<00:28,  1.92s/it]

Number of matches 23964
Number of matches After Lowe's Ratio 7835
Number of Robust matches 4357


 88%|██████████ | 107/121 [03:11<00:29,  2.10s/it]

Number of matches 23881
Number of matches After Lowe's Ratio 7588
Number of Robust matches 4685


 89%|██████████ | 108/121 [03:12<00:26,  2.05s/it]

Number of matches 22006
Number of matches After Lowe's Ratio 4746
Number of Robust matches 2624


 90%|██████████ | 109/121 [03:14<00:23,  1.96s/it]

Number of matches 21045
Number of matches After Lowe's Ratio 7802
Number of Robust matches 5716


 91%|██████████ | 110/121 [03:16<00:20,  1.86s/it]

Number of matches 21574
Number of matches After Lowe's Ratio 7636
Number of Robust matches 5313


 92%|██████████ | 111/121 [03:18<00:18,  1.85s/it]

Number of matches 19699
Number of matches After Lowe's Ratio 4815
Number of Robust matches 3387


 93%|██████████ | 112/121 [03:19<00:16,  1.78s/it]

Number of matches 20924
Number of matches After Lowe's Ratio 5048
Number of Robust matches 3304


 93%|██████████ | 113/121 [03:21<00:14,  1.77s/it]

Number of matches 21133
Number of matches After Lowe's Ratio 6789
```

```
Number of Robust matches 4103

 94%|▉▉▉▉▉▉▉▉▉▉| 114/121 [03:23<00:12,  1.72s/it]

Number of matches 21286
Number of matches After Lowe's Ratio 4963
Number of Robust matches 3245

 95%|▉▉▉▉▉▉▉▉▉▉| 115/121 [03:24<00:10,  1.75s/it]

Number of matches 20795
Number of matches After Lowe's Ratio 5513
Number of Robust matches 3642

 96%|▉▉▉▉▉▉▉▉▉▉| 116/121 [03:26<00:08,  1.71s/it]

Number of matches 20754
Number of matches After Lowe's Ratio 4846
Number of Robust matches 2135

 97%|▉▉▉▉▉▉▉▉▉▉| 117/121 [03:28<00:06,  1.67s/it]

Number of matches 21466
Number of matches After Lowe's Ratio 6528
Number of Robust matches 3355

 98%|▉▉▉▉▉▉▉▉▉▉| 118/121 [03:29<00:05,  1.68s/it]

Number of matches 22210
Number of matches After Lowe's Ratio 4487
Number of Robust matches 1720

 98%|▉▉▉▉▉▉▉▉▉▉| 119/121 [03:32<00:03,  1.82s/it]

Number of matches 22050
Number of matches After Lowe's Ratio 5959
Number of Robust matches 2655

 99%|▉▉▉▉▉▉▉▉▉▉| 120/121 [03:33<00:01,  1.78s/it]
  0%|          | 0/79 [00:00<?, ?it/s]

Number of matches 19110
Number of matches After Lowe's Ratio 1821
Number of Robust matches 551

  1%|          | 1/79 [00:00<01:17,  1.00it/s]

Number of matches 16818
Number of matches After Lowe's Ratio 3849
Number of Robust matches 1985

  3%|          | 2/79 [00:02<01:29,  1.16s/it]

Number of matches 20162
Number of matches After Lowe's Ratio 4721
Number of Robust matches 2593
```

```
  4%|▌          | 3/79 [00:04<01:48,  1.43s/it]
Number of matches 22212
Number of matches After Lowe's Ratio 4051
Number of Robust matches 2044


  5%|▌          | 4/79 [00:05<01:56,  1.56s/it]
Number of matches 21418
Number of matches After Lowe's Ratio 5235
Number of Robust matches 3647


  6%|▌          | 5/79 [00:07<01:59,  1.61s/it]
Number of matches 22258
Number of matches After Lowe's Ratio 5279
Number of Robust matches 3529


  8%|▌          | 6/79 [00:09<02:19,  1.91s/it]
Number of matches 22032
Number of matches After Lowe's Ratio 4560
Number of Robust matches 3176


  9%|▌          | 7/79 [00:11<02:17,  1.91s/it]
Number of matches 21853
Number of matches After Lowe's Ratio 7599
Number of Robust matches 5212


 10%|▌          | 8/79 [00:13<02:11,  1.86s/it]
Number of matches 23204
Number of matches After Lowe's Ratio 5709
Number of Robust matches 3063


 11%|▌          | 9/79 [00:15<02:09,  1.85s/it]
Number of matches 23002
Number of matches After Lowe's Ratio 7652
Number of Robust matches 3455


Number of matches 23539
Number of matches After Lowe's Ratio 5457
 13%|▌          | 10/79 [00:17<02:11,  1.90s/it]
Number of Robust matches 2088


 14%|▌          | 11/79 [00:19<02:14,  1.97s/it]
Number of matches 23627
Number of matches After Lowe's Ratio 6655
Number of Robust matches 2372


 15%|▌          | 12/79 [00:21<02:09,  1.94s/it]
```

```
Number of matches 22561
Number of matches After Lowe's Ratio 6815
Number of Robust matches 2457
```

```
 16%|          | 13/79 [00:23<02:04,  1.89s/it]
```

```
Number of matches 21591
Number of matches After Lowe's Ratio 6595
Number of Robust matches 2311
```

```
 18%|          | 14/79 [00:25<02:01,  1.87s/it]
```

```
Number of matches 21493
Number of matches After Lowe's Ratio 4228
Number of Robust matches 1616
```

```
 19%|          | 15/79 [00:26<01:55,  1.81s/it]
```

```
Number of matches 24774
Number of matches After Lowe's Ratio 4490
Number of Robust matches 1706
```

```
 20%|          | 16/79 [00:28<01:57,  1.86s/it]
```

```
Number of matches 26448
Number of matches After Lowe's Ratio 4859
Number of Robust matches 2061
```

```
 22%|          | 17/79 [00:31<02:08,  2.07s/it]
```

```
Number of matches 27070
Number of matches After Lowe's Ratio 4611
Number of Robust matches 1677
```

```
 23%|          | 18/79 [00:33<02:11,  2.15s/it]
```

```
Number of matches 29954
Number of matches After Lowe's Ratio 3745
Number of Robust matches 1175
```

```
 24%|          | 19/79 [00:36<02:17,  2.28s/it]
```

```
Number of matches 30498
Number of matches After Lowe's Ratio 4197
Number of Robust matches 1286
```

```
 25%|          | 20/79 [00:39<02:23,  2.44s/it]
```

```
Number of matches 30906
Number of matches After Lowe's Ratio 794
Number of Robust matches 218
```

```
 27%|          | 21/79 [00:42<02:37,  2.71s/it]
```

```
Number of matches 29804
Number of matches After Lowe's Ratio 2031
Number of Robust matches 744
```

```
 28%|[redacted]        | 22/79 [00:44<02:30,  2.65s/it]

Number of matches 28814
Number of matches After Lowe's Ratio 4971
Number of Robust matches 2473


 29%|[redacted]        | 23/79 [00:47<02:25,  2.60s/it]

Number of matches 24704
Number of matches After Lowe's Ratio 4375
Number of Robust matches 2465


 30%|[redacted]        | 24/79 [00:49<02:12,  2.40s/it]

Number of matches 24766
Number of matches After Lowe's Ratio 6538
Number of Robust matches 3417


 32%|[redacted]        | 25/79 [00:51<02:02,  2.27s/it]

Number of matches 25065
Number of matches After Lowe's Ratio 4792
Number of Robust matches 2436


 33%|[redacted]        | 26/79 [00:53<02:00,  2.28s/it]

Number of matches 26448
Number of matches After Lowe's Ratio 6526
Number of Robust matches 2609


 34%|[redacted]        | 27/79 [00:55<01:58,  2.28s/it]

Number of matches 25431
Number of matches After Lowe's Ratio 6588
Number of Robust matches 2268


 35%|[redacted]        | 28/79 [00:57<01:53,  2.23s/it]

Number of matches 26421
Number of matches After Lowe's Ratio 8453
Number of Robust matches 2639


 37%|[redacted]        | 29/79 [01:00<01:50,  2.21s/it]

Number of matches 26203
Number of matches After Lowe's Ratio 5163
Number of Robust matches 1421


 38%|[redacted]        | 30/79 [01:02<01:48,  2.22s/it]

Number of matches 24953
Number of matches After Lowe's Ratio 5380
Number of Robust matches 2402


 39%|[redacted]        | 31/79 [01:04<01:46,  2.22s/it]
```

```
Number of matches 23096
Number of matches After Lowe's Ratio 4299
Number of Robust matches 2568


 41%|[redacted]      | 32/79 [01:06<01:38,  2.09s/it]

Number of matches 23100
Number of matches After Lowe's Ratio 7478
Number of Robust matches 3932


 42%|[redacted]      | 33/79 [01:08<01:35,  2.07s/it]

Number of matches 24606
Number of matches After Lowe's Ratio 7099
Number of Robust matches 3990


 43%|[redacted]      | 34/79 [01:10<01:32,  2.04s/it]

Number of matches 24775
Number of matches After Lowe's Ratio 8699
Number of Robust matches 4182


 44%|[redacted]      | 35/79 [01:12<01:29,  2.03s/it]

Number of matches 25707
Number of matches After Lowe's Ratio 6612
Number of Robust matches 4016


 46%|[redacted]      | 36/79 [01:14<01:33,  2.17s/it]

Number of matches 25233
Number of matches After Lowe's Ratio 7870
Number of Robust matches 4814


 47%|[redacted]      | 37/79 [01:17<01:31,  2.19s/it]

Number of matches 21226
Number of matches After Lowe's Ratio 5063
Number of Robust matches 2735


 48%|[redacted]      | 38/79 [01:18<01:24,  2.07s/it]

Number of matches 22155
Number of matches After Lowe's Ratio 6476
Number of Robust matches 3785


 49%|[redacted]      | 39/79 [01:20<01:18,  1.97s/it]

Number of matches 21583
Number of matches After Lowe's Ratio 4689
Number of Robust matches 2860


 51%|[redacted]      | 40/79 [01:22<01:16,  1.95s/it]

Number of matches 22835
Number of matches After Lowe's Ratio 5910
Number of Robust matches 4218
```

```
Number of Robust matches 1210

52%|▌▌▌▌▌▌      | 41/79 [01:24<01:12,  1.90s/it]

Number of matches 23685
Number of matches After Lowe's Ratio 4600
Number of Robust matches 3069

53%|▌▌▌▌▌▌      | 42/79 [01:26<01:13,  1.99s/it]

Number of matches 23214
Number of matches After Lowe's Ratio 7055
Number of Robust matches 3826

54%|▌▌▌▌▌▌      | 43/79 [01:28<01:11,  1.98s/it]

Number of matches 21789
Number of matches After Lowe's Ratio 5882
Number of Robust matches 3344

56%|▌▌▌▌▌▌      | 44/79 [01:30<01:06,  1.91s/it]

Number of matches 24467
Number of matches After Lowe's Ratio 5282
Number of Robust matches 2894

57%|▌▌▌▌▌▌      | 45/79 [01:32<01:05,  1.93s/it]

Number of matches 26452
Number of matches After Lowe's Ratio 6556
Number of Robust matches 2718

58%|▌▌▌▌▌▌      | 46/79 [01:34<01:07,  2.05s/it]

Number of matches 27068
Number of matches After Lowe's Ratio 7263
Number of Robust matches 3623

59%|▌▌▌▌▌▌      | 47/79 [01:37<01:09,  2.18s/it]

Number of matches 25988
Number of matches After Lowe's Ratio 6788
Number of Robust matches 4918

61%|▌▌▌▌▌▌      | 48/79 [01:39<01:06,  2.14s/it]

Number of matches 23671
Number of matches After Lowe's Ratio 2126
Number of Robust matches 1120

62%|▌▌▌▌▌▌      | 49/79 [01:40<01:01,  2.04s/it]

Number of matches 22815
Number of matches After Lowe's Ratio 5518
Number of Robust matches 3262

63%|▌▌▌▌▌▌      | 50/79 [01:42<00:57,  2.00s/it]
```

```
63%|[redacted]| 50/79 [01:42<00:57,  2.00s/it]

Number of matches 21798
Number of matches After Lowe's Ratio 5313
Number of Robust matches 2694


65%|[redacted]| 51/79 [01:44<00:53,  1.89s/it]

Number of matches 19649
Number of matches After Lowe's Ratio 3757
Number of Robust matches 1940


66%|[redacted]| 52/79 [01:45<00:47,  1.78s/it]

Number of matches 19397
Number of matches After Lowe's Ratio 4939
Number of Robust matches 3034


67%|[redacted]| 53/79 [01:47<00:45,  1.76s/it]

Number of matches 19184
Number of matches After Lowe's Ratio 3081
Number of Robust matches 1893


68%|[redacted]| 54/79 [01:49<00:42,  1.72s/it]

Number of matches 19507
Number of matches After Lowe's Ratio 5356
Number of Robust matches 3603


70%|[redacted]| 55/79 [01:50<00:39,  1.66s/it]

Number of matches 20163
Number of matches After Lowe's Ratio 5817
Number of Robust matches 3669


71%|[redacted]| 56/79 [01:52<00:38,  1.66s/it]

Number of matches 22280
Number of matches After Lowe's Ratio 5818
Number of Robust matches 4310


72%|[redacted]| 57/79 [01:54<00:37,  1.71s/it]

Number of matches 25286
Number of matches After Lowe's Ratio 7310
Number of Robust matches 5136


73%|[redacted]| 58/79 [01:56<00:41,  2.00s/it]

Number of matches 26666
Number of matches After Lowe's Ratio 7020
Number of Robust matches 5254


75%|[redacted]| 59/79 [01:59<00:42,  2.13s/it]

Number of matches 26685
Number of matches After Lowe's Ratio 7704
```

```
Number of Robust matches 5214



Number of matches 24536
Number of matches After Lowe's Ratio 6762
 77%|[redacted]  | 61/79 [02:03<00:37,  2.10s/it]
Number of Robust matches 4452



Number of matches 22328
Number of matches After Lowe's Ratio 7682
Number of Robust matches 4787


 78%|[redacted]  | 62/79 [02:05<00:34,  2.01s/it]
Number of matches 24173
Number of matches After Lowe's Ratio 7949
Number of Robust matches 4729


 80%|[redacted]  | 63/79 [02:07<00:31,  1.99s/it]
Number of matches 25542
Number of matches After Lowe's Ratio 8655
Number of Robust matches 4603


 81%|[redacted]  | 64/79 [02:09<00:31,  2.13s/it]
Number of matches 25779
Number of matches After Lowe's Ratio 9347
Number of Robust matches 4358


 82%|[redacted]  | 65/79 [02:11<00:29,  2.12s/it]
Number of matches 23910
Number of matches After Lowe's Ratio 6724
Number of Robust matches 2733


 84%|[redacted]  | 66/79 [02:13<00:26,  2.07s/it]
Number of matches 22719
Number of matches After Lowe's Ratio 6734
Number of Robust matches 2609


 85%|[redacted]  | 67/79 [02:15<00:24,  2.04s/it]
Number of matches 24382
Number of matches After Lowe's Ratio 6530
Number of Robust matches 2296


 86%|[redacted]  | 68/79 [02:17<00:22,  2.01s/it]
Number of matches 23709
Number of matches After Lowe's Ratio 7593
Number of Robust matches 3320


 87%|[redacted]  | 69/79 [02:19<00:19,  1.97s/it]
```

```
Number of matches 23640
Number of matches After Lowe's Ratio 6047
Number of Robust matches 2663


Number of matches 24162
Number of matches After Lowe's Ratio 4642
89%|[redacted] | 70/79 [02:21<00:18,  2.07s/it]
Number of Robust matches 2458


90%|[redacted] | 71/79 [02:23<00:16,  2.02s/it]
Number of matches 24783
Number of matches After Lowe's Ratio 5123
Number of Robust matches 2500


91%|[redacted] | 72/79 [02:25<00:14,  2.01s/it]
Number of matches 25360
Number of matches After Lowe's Ratio 4447
Number of Robust matches 2232


92%|[redacted] | 73/79 [02:27<00:12,  2.03s/it]
Number of matches 25982
Number of matches After Lowe's Ratio 4114
Number of Robust matches 1745


94%|[redacted] | 74/79 [02:30<00:10,  2.12s/it]
Number of matches 29135
Number of matches After Lowe's Ratio 3959
Number of Robust matches 1662


95%|[redacted] | 75/79 [02:33<00:09,  2.46s/it]
Number of matches 30232
Number of matches After Lowe's Ratio 4243
Number of Robust matches 2038


96%|[redacted] | 76/79 [02:35<00:07,  2.47s/it]
Number of matches 29257
Number of matches After Lowe's Ratio 682
Number of Robust matches 180


97%|[redacted] | 77/79 [02:38<00:04,  2.50s/it]
Number of matches 26048
Number of matches After Lowe's Ratio 3532
Number of Robust matches 1760


99%|[redacted] | 78/79 [02:40<00:02,  2.06s/it]
Number of matches 25271
```

```
Number of Matches After Lowe's Ratio 5909
Number of Robust matches 3189
```

In [ ]:

```
H_left_freak = []
H_right_freak = []

num_matches_freak = []
num_good_matches_freak = []

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

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

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

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

In [ ]:

```
H_left_mser = []
H_right_mser = []

num_matches_mser = []
num_good_matches_mser = []

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

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

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

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

In [ ]:

```
H_left_gftt = []
H_right_gftt = []

num_matches_gftt = []
```

```
num_good_matches_gftt = []

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

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

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

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

In [ ]:

```
H_left_daisy = []
H_right_daisy = []

num_matches_daisy = []
num_good_matches_daisy = []

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

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

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

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

In [ ]:

```
H_left_fast = []
H_right_fast = []

num_matches_fast = []
num_good_matches_fast = []

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

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

```
    num_good_matches_fast.append(gd_matches)

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

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

In [ ]:

```
H_left_star = []
H_right_star = []

num_matches_star = []
num_good_matches_star = []

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

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_star[j:j+2][::-1],points_all_left_star[j:j+2][::-1],descriptors_all_left_brief[j:j+2][:
:-1])
    H_left_star.append(H_a)
    num_matches_star.append(matches)
    num_good_matches_star.append(gd_matches)

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

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_star[j:j+2][::-1],points_all_right_star[j:j+2][::-1],descriptors_all_right_brief[j:j+
2][::-1])
    H_right_star.append(H_a)
    num_matches_star.append(matches)
    num_good_matches_star.append(gd_matches)
```

In [ ]:

```
H_left_sift = []
H_right_sift = []

num_matches_sift = []
num_good_matches_sift = []

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

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_sift[j:j+2][::-1],points_all_left_sift[j:j+2][::-1],descriptors_all_left_sift[j:j+2][::
-1])
    H_left_sift.append(H_a)
    num_matches_sift.append(matches)
    num_good_matches_sift.append(gd_matches)

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

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_sift[j:j+2][::-1],points_all_right_sift[j:j+2][::-1],descriptors_all_right_sift[j:j+2
][::-1])
    H_right_sift.append(H_a)
    num_matches_sift.append(matches)
```

```
    num_good_matches_sift.append(gd_matches)
```

In [ ]:

```
H_left_surf = []
H_right_surf = []

num_matches_surf = []
num_good_matches_surf = []

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

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_surf[j:j+2][::-1],points_all_left_surf[j:j++2][::-1],descriptors_all_left_surf[j:j+2][:
:-1])
    H_left_surf.append(H_a)
    num_matches_surf.append(matches)
    num_good_matches_surf.append(gd_matches)

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

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_surf[j:j+2][::-1],points_all_right_surf[j:j+2][::-1],descriptors_all_right_surf[j:j+2
][::-1])
    H_right_surf.append(H_a)
    num_matches_surf.append(matches)
    num_good_matches_surf.append(gd_matches)
```

In [21]:

```
H_left_surfsift = []
H_right_surfsift = []

num_matches_surfsift = []
num_good_matches_surfsift = []

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

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_surfsift[j:j+2][::-1],points_all_left_surfsift[j:j++2][::-1],descriptors_all_left_surfs
ift[j:j+2][::-1])
    H_left_surfsift.append(H_a)
    num_matches_surfsift.append(matches)
    num_good_matches_surfsift.append(gd_matches)

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

    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_surfsift[j:j+2][::-1],points_all_right_surfsift[j:j+2][::-1],descriptors_all_right_su
rfsift[j:j+2][::-1])
    H_right_surfsift.append(H_a)
    num_matches_surfsift.append(matches)
    num_good_matches_surfsift.append(gd_matches)
```

```
  1%|          | 1/121 [00:01<03:35,  1.80s/it]
```

```
Number of matches 14781
Number of matches After Lowe's Ratio 4556
Number of Robust matches 2280
```

```
  2%|          | 2/121 [00:03<03:30,  1.77s/it]
```

```
Number of matches 16801
```

```
Number of matches 16801
Number of matches After Lowe's Ratio 4621
Number of Robust matches 2526


  2%|          | 3/121 [00:05<03:59,  2.03s/it]

Number of matches 20090
Number of matches After Lowe's Ratio 5172
Number of Robust matches 2342


  3%|          | 4/121 [00:08<04:32,  2.33s/it]

Number of matches 25235
Number of matches After Lowe's Ratio 5823
Number of Robust matches 2748


  4%|          | 5/121 [00:12<05:36,  2.90s/it]

Number of matches 28279
Number of matches After Lowe's Ratio 3639
Number of Robust matches 1564


  5%|          | 6/121 [00:16<06:19,  3.30s/it]

Number of matches 28533
Number of matches After Lowe's Ratio 5891
Number of Robust matches 2045


  6%|          | 7/121 [00:21<06:58,  3.67s/it]

Number of matches 29756
Number of matches After Lowe's Ratio 3501
Number of Robust matches 1627


  7%|          | 8/121 [00:25<07:29,  3.98s/it]

Number of matches 32962
Number of matches After Lowe's Ratio 1384
Number of Robust matches 391


  7%|          | 9/121 [00:30<07:50,  4.21s/it]

Number of matches 30289
Number of matches After Lowe's Ratio 4711
Number of Robust matches 2091


Number of matches 28235
Number of matches After Lowe's Ratio 5895

  8%|          | 10/121 [00:34<07:55,  4.28s/it]

Number of Robust matches 2727


  9%|          | 11/121 [00:38<07:39,  4.18s/it]

Number of matches 26562
Number of matches After Lowe's Ratio 5230
Number of Robust matches 2412
```

```
 10%|          | 12/121 [00:42<07:17,  4.02s/it]
```

```
Number of matches 27072
Number of matches After Lowe's Ratio 5969
Number of Robust matches 2951
```

```
 11%|          | 13/121 [00:46<06:58,  3.88s/it]
```

```
Number of matches 23782
Number of matches After Lowe's Ratio 7198
Number of Robust matches 3655
```

```
 12%|          | 14/121 [00:48<06:25,  3.60s/it]
```

```
Number of matches 16085
Number of matches After Lowe's Ratio 3276
Number of Robust matches 1497
```

```
 12%|          | 15/121 [00:50<05:24,  3.06s/it]
```

```
Number of matches 14514
Number of matches After Lowe's Ratio 4476
Number of Robust matches 1919
```

```
 13%|          | 16/121 [00:52<04:36,  2.63s/it]
```

```
Number of matches 14985
Number of matches After Lowe's Ratio 2332
Number of Robust matches 875
```

```
 14%|          | 17/121 [00:54<04:08,  2.39s/it]
```

```
Number of matches 18671
Number of matches After Lowe's Ratio 4348
Number of Robust matches 1324
```

```
 15%|          | 18/121 [00:56<04:13,  2.46s/it]
```

```
Number of matches 27675
Number of matches After Lowe's Ratio 2625
Number of Robust matches 1326
```

```
 16%|          | 19/121 [01:01<05:05,  3.00s/it]
```

```
Number of matches 27970
Number of matches After Lowe's Ratio 10085
Number of Robust matches 4552
```

```
 17%|          | 20/121 [01:05<05:31,  3.29s/it]
```

```
Number of matches 25077
Number of matches After Lowe's Ratio 5335
Number of Robust matches 2462
```

```
 17%|          | 21/121 [01:08<05:38,  3.38s/it]
```

```
Number of matches 26661
Number of matches After Lowe's Ratio 8107
Number of Robust matches 3419
```

```
 18%|          | 22/121 [01:13<06:04,  3.68s/it]
```

```
Number of matches 27835
Number of matches After Lowe's Ratio 5115
Number of Robust matches 2232
```

```
 19%|          | 23/121 [01:17<06:11,  3.80s/it]
```

```
Number of matches 26784
Number of matches After Lowe's Ratio 4954
Number of Robust matches 2399
```

```
 20%|          | 24/121 [01:20<06:07,  3.79s/it]
```

```
Number of matches 26930
Number of matches After Lowe's Ratio 3625
Number of Robust matches 1821
```

```
 21%|          | 25/121 [01:24<06:11,  3.87s/it]
```

```
Number of matches 28008
Number of matches After Lowe's Ratio 4699
Number of Robust matches 2038
```

```
 23%|          | 28/121 [01:37<06:31,  4.21s/it]
```

```
Number of matches 32633
Number of matches After Lowe's Ratio 5346
Number of Robust matches 1811
```

```
 24%|          | 29/121 [01:42<06:49,  4.45s/it]
```

```
Number of matches 33122
Number of matches After Lowe's Ratio 9289
Number of Robust matches 2723
```

```
 25%|          | 30/121 [01:48<07:08,  4.71s/it]
```

```
Number of matches 32873
Number of matches After Lowe's Ratio 5423
Number of Robust matches 1524
```

```
 26%|          | 31/121 [01:52<07:04,  4.71s/it]
```

```
Number of matches 28706
Number of matches After Lowe's Ratio 5022
Number of Robust matches 1466
```

```
 26%|          | 32/121 [01:57<06:42,  4.53s/it]
```

```
Number of matches 28350
Number of matches After Lowe's Ratio 5367
Number of Robust matches 1818
```

```
Number of Robust matches 1010

 27%|██▋       | 33/121 [02:01<06:34,  4.48s/it]

Number of matches 29422
Number of matches After Lowe's Ratio 6004
Number of Robust matches 2026

 28%|██▊       | 34/121 [02:05<06:27,  4.45s/it]

Number of matches 31567
Number of matches After Lowe's Ratio 7103
Number of Robust matches 2197

 29%|██▉       | 35/121 [02:10<06:34,  4.59s/it]

Number of matches 34036
Number of matches After Lowe's Ratio 3489
Number of Robust matches 972

 30%|███       | 36/121 [02:15<06:43,  4.75s/it]

Number of matches 30610
Number of matches After Lowe's Ratio 78
Number of Robust matches 14

 31%|███       | 37/121 [02:20<06:36,  4.72s/it]

Number of matches 32011
Number of matches After Lowe's Ratio 3885
Number of Robust matches 1039

 31%|▉         | 38/121 [02:24<06:19,  4.58s/it]

Number of matches 27897
Number of matches After Lowe's Ratio 1348
Number of Robust matches 385

 32%|▉         | 39/121 [02:29<06:07,  4.48s/it]

Number of matches 30089
Number of matches After Lowe's Ratio 6270
Number of Robust matches 2188

 33%|▉         | 40/121 [02:33<05:59,  4.44s/it]

Number of matches 27735
Number of matches After Lowe's Ratio 6899
Number of Robust matches 2309

 34%|▉         | 41/121 [02:37<05:53,  4.42s/it]

Number of matches 31322
Number of matches After Lowe's Ratio 7765
Number of Robust matches 2488

 35%|▉         | 42/121 [02:42<05:52,  4.48s/it]
```

```
35%|[redacted]| 42/121 [02:42<05:53,  4.48s/it]

Number of matches 32266
Number of matches After Lowe's Ratio 9359
Number of Robust matches 3419


36%|[redacted]| 43/121 [02:47<05:54,  4.54s/it]

Number of matches 31185
Number of matches After Lowe's Ratio 7191
Number of Robust matches 2654


36%|[redacted]| 44/121 [02:51<05:55,  4.61s/it]

Number of matches 30146
Number of matches After Lowe's Ratio 8333
Number of Robust matches 3132


37%|[redacted]| 45/121 [02:56<05:52,  4.63s/it]

Number of matches 31517
Number of matches After Lowe's Ratio 7942
Number of Robust matches 3336


38%|[redacted]| 46/121 [03:01<05:55,  4.74s/it]

Number of matches 30289
Number of matches After Lowe's Ratio 7866
Number of Robust matches 3339


39%|[redacted]| 47/121 [03:06<05:45,  4.67s/it]

Number of matches 29532
Number of matches After Lowe's Ratio 7264
Number of Robust matches 3182


40%|[redacted]| 48/121 [03:10<05:32,  4.56s/it]

Number of matches 27853
Number of matches After Lowe's Ratio 7809
Number of Robust matches 4049


40%|[redacted]| 49/121 [03:14<05:26,  4.53s/it]

Number of matches 28712
Number of matches After Lowe's Ratio 6043
Number of Robust matches 3334


41%|[redacted]| 50/121 [03:18<05:14,  4.43s/it]

Number of matches 28137
Number of matches After Lowe's Ratio 7846
Number of Robust matches 4990


42%|[redacted]| 51/121 [03:23<05:08,  4.40s/it]

Number of matches 32449
Number of matches After Lowe's Ratio 7636
```

```
Number of Robust matches 4982

 43%|[redacted]      | 52/121 [03:28<05:17,  4.60s/it]

Number of matches 30475
Number of matches After Lowe's Ratio 9960
Number of Robust matches 6017

 44%|[redacted]      | 53/121 [03:32<05:12,  4.60s/it]

Number of matches 32804
Number of matches After Lowe's Ratio 9939
Number of Robust matches 4385

 45%|[redacted]      | 54/121 [03:37<05:07,  4.59s/it]

Number of matches 23382
Number of matches After Lowe's Ratio 3075
Number of Robust matches 1468

 45%|[redacted]      | 55/121 [03:41<04:41,  4.27s/it]

Number of matches 25745
Number of matches After Lowe's Ratio 5350
Number of Robust matches 2724

 46%|[redacted]      | 56/121 [03:44<04:22,  4.04s/it]

Number of matches 22006
Number of matches After Lowe's Ratio 1398
Number of Robust matches 611

 47%|[redacted]      | 57/121 [03:47<04:02,  3.79s/it]

Number of matches 27439
Number of matches After Lowe's Ratio 2970
Number of Robust matches 1466

 48%|[redacted]      | 58/121 [03:51<04:07,  3.92s/it]

Number of matches 30994
Number of matches After Lowe's Ratio 5827
Number of Robust matches 2856

 49%|[redacted]      | 59/121 [03:56<04:09,  4.02s/it]

Number of matches 29394
Number of matches After Lowe's Ratio 5454
Number of Robust matches 2786

 50%|[redacted]      | 60/121 [04:00<04:08,  4.08s/it]

Number of matches 30765
Number of matches After Lowe's Ratio 3618
Number of Robust matches 1499
```

```
 50%|█████     | 61/121 [04:04<04:12,  4.21s/it]
Number of matches 29759
Number of matches After Lowe's Ratio 3190
Number of Robust matches 1036

 51%|█████     | 62/121 [04:09<04:12,  4.29s/it]
Number of matches 31803
Number of matches After Lowe's Ratio 2785
Number of Robust matches 865

 52%|█████▏    | 63/121 [04:13<04:10,  4.32s/it]
Number of matches 28470
Number of matches After Lowe's Ratio 599
Number of Robust matches 133

 53%|█████▎    | 64/121 [04:18<04:05,  4.31s/it]
Number of matches 30574
Number of matches After Lowe's Ratio 6192
Number of Robust matches 2416

 54%|█████▎    | 65/121 [04:22<04:00,  4.30s/it]
Number of matches 28292
Number of matches After Lowe's Ratio 5565
Number of Robust matches 2221

 55%|█████▍    | 66/121 [04:26<03:52,  4.23s/it]
Number of matches 28159
Number of matches After Lowe's Ratio 5429
Number of Robust matches 2118

 55%|█████▌    | 67/121 [04:30<03:49,  4.26s/it]
Number of matches 30159
Number of matches After Lowe's Ratio 5894
Number of Robust matches 1744

 56%|█████▌    | 68/121 [04:34<03:43,  4.21s/it]
Number of matches 25112
Number of matches After Lowe's Ratio 2929
Number of Robust matches 1243

 57%|█████▋    | 69/121 [04:38<03:29,  4.03s/it]
Number of matches 29686
Number of matches After Lowe's Ratio 5276
Number of Robust matches 1904

 58%|█████▊    | 70/121 [04:42<03:25,  4.04s/it]
Number of matches 23395
Number of matches After Lowe's Ratio 2734
```

```
Number of Robust matches 1102

 59%|▋▋▋▋▋▋     | 71/121 [04:45<03:12,  3.85s/it]

Number of matches 27431
Number of matches After Lowe's Ratio 5819
Number of Robust matches 2603

 60%|▋▋▋▋▋▋     | 72/121 [04:49<03:09,  3.87s/it]

Number of matches 27034
Number of matches After Lowe's Ratio 5771
Number of Robust matches 2299

 60%|▋▋▋▋▋▋     | 73/121 [04:54<03:10,  3.97s/it]

Number of matches 29867
Number of matches After Lowe's Ratio 5546
Number of Robust matches 2523

 61%|▋▋▋▋▋▋     | 74/121 [04:58<03:13,  4.12s/it]

Number of matches 31522
Number of matches After Lowe's Ratio 9253
Number of Robust matches 5344

 62%|▋▋▋▋▋▋     | 75/121 [05:03<03:18,  4.32s/it]

Number of matches 30097
Number of matches After Lowe's Ratio 6990
Number of Robust matches 3495

 63%|▋▋▋▋▋▋     | 76/121 [05:07<03:15,  4.35s/it]

Number of matches 30110
Number of matches After Lowe's Ratio 7494
Number of Robust matches 3906

 64%|▋▋▋▋▋▋     | 77/121 [05:12<03:11,  4.34s/it]

Number of matches 29782
Number of matches After Lowe's Ratio 7381
Number of Robust matches 4409

 64%|▋▋▋▋▋▋     | 78/121 [05:16<03:09,  4.42s/it]

Number of matches 29162
Number of matches After Lowe's Ratio 7774
Number of Robust matches 4348

 65%|▋▋▋▋▋▋     | 79/121 [05:21<03:05,  4.41s/it]

Number of matches 28941
Number of matches After Lowe's Ratio 8998
Number of Robust matches 5155
```

```
 66%|▉▉▉▉▉▉▉    | 80/121 [05:25<02:58,  4.34s/it]

Number of matches 29718
Number of matches After Lowe's Ratio 7700
Number of Robust matches 4117


 67%|▉▉▉▉▉▉▉▉   | 81/121 [05:29<02:56,  4.41s/it]

Number of matches 29774
Number of matches After Lowe's Ratio 6811
Number of Robust matches 3058


 68%|▉▉▉▉▉▉▉▉   | 82/121 [05:34<02:49,  4.34s/it]

Number of matches 28465
Number of matches After Lowe's Ratio 7401
Number of Robust matches 4156


Number of matches 28901
Number of matches After Lowe's Ratio 6904
 69%|▉▉▉▉▉▉▉▉   | 83/121 [05:38<02:44,  4.32s/it]

Number of Robust matches 2641


 69%|▉▉▉▉▉▉▉▉   | 84/121 [05:42<02:37,  4.27s/it]

Number of matches 29317
Number of matches After Lowe's Ratio 6403
Number of Robust matches 2632


 70%|▉▉▉▉▉▉▉▉   | 85/121 [05:46<02:33,  4.27s/it]

Number of matches 32274
Number of matches After Lowe's Ratio 4340
Number of Robust matches 1726


 71%|▉▉▉▉▉▉▉▉   | 86/121 [05:51<02:34,  4.43s/it]

Number of matches 29333
Number of matches After Lowe's Ratio 4979
Number of Robust matches 1757


 72%|▉▉▉▉▉▉▉▉▉  | 87/121 [05:55<02:29,  4.40s/it]

Number of matches 28791
Number of matches After Lowe's Ratio 5697
Number of Robust matches 2286


 73%|▉▉▉▉▉▉▉▉▉  | 88/121 [06:00<02:23,  4.35s/it]

Number of matches 31214
Number of matches After Lowe's Ratio 5698
Number of Robust matches 1476


 74%|▉▉▉▉▉▉▉▉▉  | 89/121 [06:04<02:21,  4.42s/it]

Number of matches 28464
```

```
Number of matches 28464
Number of matches After Lowe's Ratio 3310
Number of Robust matches 1121
```

```
 74%|███████▍  | 90/121 [06:08<02:13,  4.30s/it]
```

```
Number of matches 29409
Number of matches After Lowe's Ratio 5080
Number of Robust matches 2020
```

```
 75%|███████▌  | 91/121 [06:12<02:07,  4.25s/it]
```

```
Number of matches 27445
Number of matches After Lowe's Ratio 2605
Number of Robust matches 1073
```

```
 76%|███████▌  | 92/121 [06:16<02:01,  4.18s/it]
```

```
Number of matches 26763
Number of matches After Lowe's Ratio 1067
Number of Robust matches 274
```

```
 77%|███████▋  | 93/121 [06:20<01:53,  4.04s/it]
```

```
Number of matches 27227
Number of matches After Lowe's Ratio 5122
Number of Robust matches 2478
```

```
 78%|███████▊  | 94/121 [06:24<01:47,  3.97s/it]
```

```
Number of matches 26057
Number of matches After Lowe's Ratio 6117
Number of Robust matches 2863
```

```
 79%|███████▊  | 95/121 [06:28<01:44,  4.02s/it]
```

```
Number of matches 27493
Number of matches After Lowe's Ratio 5522
Number of Robust matches 2591
```

```
 79%|███████▉  | 96/121 [06:32<01:40,  4.01s/it]
```

```
Number of matches 28655
Number of matches After Lowe's Ratio 6433
Number of Robust matches 2348
```

```
 80%|████████  | 97/121 [06:36<01:37,  4.05s/it]
```

```
Number of matches 31224
Number of matches After Lowe's Ratio 4800
Number of Robust matches 1878
```

```
 81%|████████  | 98/121 [06:41<01:38,  4.27s/it]
```

```
Number of matches 32243
Number of matches After Lowe's Ratio 5600
Number of Robust matches 1565
```

```
 82%|▉▉▉▉▉▉▉▉▉▉  | 99/121 [06:46<01:36,  4.38s/it]

Number of matches 31404
Number of matches After Lowe's Ratio 5026
Number of Robust matches 1812


 83%|▉▉▉▉▉▉▉▉▉▉  | 100/121 [06:50<01:33,  4.46s/it]

Number of matches 29941
Number of matches After Lowe's Ratio 6498
Number of Robust matches 2415


 83%|▉▉▉▉▉▉▉▉▉▉  | 101/121 [06:54<01:27,  4.39s/it]

Number of matches 29731
Number of matches After Lowe's Ratio 6816
Number of Robust matches 3522


 84%|▉▉▉▉▉▉▉▉▉▉  | 102/121 [06:59<01:22,  4.36s/it]

Number of matches 29003
Number of matches After Lowe's Ratio 7234
Number of Robust matches 3855


 85%|▉▉▉▉▉▉▉▉▉▉  | 103/121 [07:03<01:18,  4.35s/it]

Number of matches 29540
Number of matches After Lowe's Ratio 8540
Number of Robust matches 4353


 86%|▉▉▉▉▉▉▉▉▉▉  | 104/121 [07:07<01:13,  4.34s/it]

Number of matches 30714
Number of matches After Lowe's Ratio 7688
Number of Robust matches 4043


 87%|▉▉▉▉▉▉▉▉▉▉  | 105/121 [07:12<01:10,  4.40s/it]

Number of matches 31999
Number of matches After Lowe's Ratio 8209
Number of Robust matches 3966


 88%|▉▉▉▉▉▉▉▉▉▉  | 106/121 [07:17<01:08,  4.56s/it]

Number of matches 31349
Number of matches After Lowe's Ratio 8523
Number of Robust matches 4475


 88%|▉▉▉▉▉▉▉▉▉▉  | 107/121 [07:22<01:04,  4.61s/it]

Number of matches 31491
Number of matches After Lowe's Ratio 7657
Number of Robust matches 3401


 89%|▉▉▉▉▉▉▉▉▉▉  | 108/121 [07:26<00:59,  4.59s/it]
```

```
Number of matches 29793
Number of matches After Lowe's Ratio 4676
Number of Robust matches 1975


 90%|██████████ | 109/121 [07:30<00:53,  4.49s/it]

Number of matches 28686
Number of matches After Lowe's Ratio 8571
Number of Robust matches 4652


 91%|██████████ | 110/121 [07:34<00:48,  4.37s/it]

Number of matches 29518
Number of matches After Lowe's Ratio 8902
Number of Robust matches 4655


 92%|██████████ | 111/121 [07:39<00:43,  4.39s/it]

Number of matches 27220
Number of matches After Lowe's Ratio 5357
Number of Robust matches 3076


 93%|██████████ | 112/121 [07:43<00:37,  4.21s/it]

Number of matches 27631
Number of matches After Lowe's Ratio 5598
Number of Robust matches 2841


 93%|██████████ | 113/121 [07:47<00:32,  4.11s/it]

Number of matches 28608
Number of matches After Lowe's Ratio 7335
Number of Robust matches 3839


 94%|██████████ | 114/121 [07:51<00:28,  4.11s/it]

Number of matches 27793
Number of matches After Lowe's Ratio 5386
Number of Robust matches 2493


 95%|██████████ | 115/121 [07:54<00:24,  4.01s/it]

Number of matches 26647
Number of matches After Lowe's Ratio 5632
Number of Robust matches 2286



Number of matches 27094
Number of matches After Lowe's Ratio 5100
 96%|██████████ | 116/121 [07:58<00:19,  3.95s/it]

Number of Robust matches 2132


 97%|██████████ | 117/121 [08:02<00:15,  3.90s/it]

Number of matches 28742
Number of matches After Lowe's Ratio 7371
Number of Robust matches 2736
```

```
Number of Robust matches 2750

 98%|██████████| 118/121 [08:06<00:11,  3.95s/it]

Number of matches 28988
Number of matches After Lowe's Ratio 4197
Number of Robust matches 1156

 98%|██████████| 119/121 [08:11<00:08,  4.12s/it]

Number of matches 29103
Number of matches After Lowe's Ratio 6526
Number of Robust matches 1954

 99%|██████████| 120/121 [08:15<00:04,  4.13s/it]
  0%|          | 0/79 [00:00<?, ?it/s]

Number of matches 28791
Number of matches After Lowe's Ratio 1693
Number of Robust matches 404

  1%|          | 1/79 [00:02<03:02,  2.34s/it]

Number of matches 21194
Number of matches After Lowe's Ratio 5110
Number of Robust matches 2092

Number of matches 25233
Number of matches After Lowe's Ratio 6513

  3%|          | 2/79 [00:05<03:34,  2.78s/it]

Number of Robust matches 2697

  4%|          | 3/79 [00:08<03:57,  3.12s/it]

Number of matches 26393
Number of matches After Lowe's Ratio 4655
Number of Robust matches 2054

  5%|          | 4/79 [00:12<04:09,  3.32s/it]

Number of matches 25164
Number of matches After Lowe's Ratio 6700
Number of Robust matches 3647

  6%|          | 5/79 [00:16<04:11,  3.39s/it]

Number of matches 25192
Number of matches After Lowe's Ratio 6566
Number of Robust matches 3938

  8%|          | 6/79 [00:19<04:17,  3.53s/it]

Number of matches 26573
Number of matches After Lowe's Ratio 5301
Number of Robust matches 3024
```

```
  9%|          | 7/79 [00:23<04:20,  3.61s/it]
Number of matches 29677
Number of matches After Lowe's Ratio 8990
Number of Robust matches 5150


Number of matches 32162
Number of matches After Lowe's Ratio 7129
 10%|          | 8/79 [00:28<04:39,  3.93s/it]
Number of Robust matches 2684


 11%|          | 9/79 [00:33<04:55,  4.23s/it]
Number of matches 33265
Number of matches After Lowe's Ratio 10079
Number of Robust matches 3219


 13%|          | 10/79 [00:38<05:04,  4.41s/it]
Number of matches 32297
Number of matches After Lowe's Ratio 5790
Number of Robust matches 1757


 14%|          | 11/79 [00:43<05:13,  4.61s/it]
Number of matches 31723
Number of matches After Lowe's Ratio 7062
Number of Robust matches 1948


 15%|          | 12/79 [00:47<05:10,  4.63s/it]
Number of matches 33216
Number of matches After Lowe's Ratio 7881
Number of Robust matches 2368


Number of matches 28873
Number of matches After Lowe's Ratio 6957
 16%|          | 13/79 [00:52<05:12,  4.73s/it]
Number of Robust matches 1971


 18%|          | 14/79 [00:56<04:54,  4.52s/it]
Number of matches 28935
Number of matches After Lowe's Ratio 5586
Number of Robust matches 1898


 19%|          | 15/79 [01:01<04:47,  4.49s/it]
Number of matches 30247
Number of matches After Lowe's Ratio 5712
Number of Robust matches 1857
```

```
 20%|██        | 16/79 [01:05<04:45,  4.54s/it]
```

```
Number of matches 30604
Number of matches After Lowe's Ratio 5766
Number of Robust matches 2036
```

```
 22%|██        | 17/79 [01:10<04:41,  4.54s/it]
```

```
Number of matches 31258
Number of matches After Lowe's Ratio 5458
Number of Robust matches 1755
```

```
 23%|██        | 18/79 [01:15<04:41,  4.61s/it]
```

```
Number of matches 33741
Number of matches After Lowe's Ratio 3915
Number of Robust matches 1218
```

```
 24%|██        | 19/79 [01:20<04:47,  4.80s/it]
```

```
Number of matches 33862
Number of matches After Lowe's Ratio 4914
Number of Robust matches 1400
```

```
 25%|██        | 20/79 [01:25<04:47,  4.87s/it]
```

```
Number of matches 34011
Number of matches After Lowe's Ratio 806
Number of Robust matches 289
```

```
 27%|██        | 21/79 [01:30<04:46,  4.93s/it]
```

```
Number of matches 33052
Number of matches After Lowe's Ratio 1910
Number of Robust matches 660
```

```
 28%|██        | 22/79 [01:35<04:44,  5.00s/it]
```

```
Number of matches 32595
Number of matches After Lowe's Ratio 5756
Number of Robust matches 2125
```

```
 29%|██        | 23/79 [01:40<04:31,  4.86s/it]
```

```
Number of matches 32188
Number of matches After Lowe's Ratio 4444
Number of Robust matches 1777
```

```
 30%|└└        | 24/79 [01:45<04:28,  4.88s/it]
```

```
Number of matches 33091
Number of matches After Lowe's Ratio 8477
Number of Robust matches 3569
```

```
 32%|└└└       | 25/79 [01:49<04:22,  4.86s/it]
```

```
Number of matches 34266
Number of matches After Lowe's Ratio 5988
```

```
Number of Robust matches 2466

 33%|▉▉▉       | 26/79 [01:55<04:29,  5.08s/it]

Number of matches 37345
Number of matches After Lowe's Ratio 8190
Number of Robust matches 2298

 34%|▉▉▉       | 27/79 [02:00<04:30,  5.21s/it]

Number of matches 32231
Number of matches After Lowe's Ratio 6125
Number of Robust matches 1671

 35%|▉▉▉       | 28/79 [02:06<04:24,  5.18s/it]

Number of matches 33997
Number of matches After Lowe's Ratio 8998
Number of Robust matches 2014

 37%|▉▉▉       | 29/79 [02:11<04:20,  5.21s/it]

Number of matches 33681
Number of matches After Lowe's Ratio 5023
Number of Robust matches 1381

 38%|▉▉▉       | 30/79 [02:16<04:12,  5.16s/it]

Number of matches 31387
Number of matches After Lowe's Ratio 5363
Number of Robust matches 2015

 39%|▉▉▉       | 31/79 [02:21<04:03,  5.07s/it]

Number of matches 30674
Number of matches After Lowe's Ratio 5051
Number of Robust matches 2447

 41%|▉▉▉▉      | 32/79 [02:25<03:52,  4.95s/it]

Number of matches 30596
Number of matches After Lowe's Ratio 8726
Number of Robust matches 3501

 42%|▉▉▉▉      | 33/79 [02:30<03:42,  4.83s/it]

Number of matches 31084
Number of matches After Lowe's Ratio 8464
Number of Robust matches 3689

 43%|▉▉▉▉      | 34/79 [02:35<03:36,  4.80s/it]

Number of matches 31253
Number of matches After Lowe's Ratio 10360
Number of Robust matches 4457
```

```
 44%|[redacted]      | 35/79 [02:39<03:29,  4.77s/it]

Number of matches 30841
Number of matches After Lowe's Ratio 7261
Number of Robust matches 3399


 46%|[redacted]      | 36/79 [02:44<03:22,  4.71s/it]

Number of matches 30399
Number of matches After Lowe's Ratio 8849
Number of Robust matches 4080


 47%|[redacted]      | 37/79 [02:48<03:13,  4.60s/it]

Number of matches 26077
Number of matches After Lowe's Ratio 5324
Number of Robust matches 1846


 48%|[redacted]      | 38/79 [02:52<02:57,  4.34s/it]

Number of matches 27224
Number of matches After Lowe's Ratio 6565
Number of Robust matches 2916


 49%|[redacted]      | 39/79 [02:56<02:45,  4.13s/it]

Number of matches 26927
Number of matches After Lowe's Ratio 5028
Number of Robust matches 2336


 51%|[redacted]      | 40/79 [03:00<02:39,  4.08s/it]

Number of matches 28259
Number of matches After Lowe's Ratio 6388
Number of Robust matches 2730


 52%|[redacted]      | 41/79 [03:03<02:32,  4.00s/it]

Number of matches 27495
Number of matches After Lowe's Ratio 5109
Number of Robust matches 2920



Number of matches 27753
Number of matches After Lowe's Ratio 7216
 53%|[redacted]      | 42/79 [03:08<02:28,  4.01s/it]

Number of Robust matches 3475


 54%|[redacted]      | 43/79 [03:11<02:22,  3.97s/it]

Number of matches 28990
Number of matches After Lowe's Ratio 6215
Number of Robust matches 3132


 56%|[redacted]      | 44/79 [03:16<02:23,  4.11s/it]

Number of matches 29470
```

```
Number of matches 29470
Number of matches After Lowe's Ratio 5170
Number of Robust matches 2090

 57%|[redacted]     | 45/79 [03:20<02:23,  4.22s/it]

Number of matches 31174
Number of matches After Lowe's Ratio 6411
Number of Robust matches 2302

 58%|[redacted]     | 46/79 [03:25<02:23,  4.36s/it]

Number of matches 32280
Number of matches After Lowe's Ratio 6972
Number of Robust matches 2419

 59%|[redacted]     | 47/79 [03:30<02:23,  4.48s/it]

Number of matches 30821
Number of matches After Lowe's Ratio 6777
Number of Robust matches 3462

 61%|[redacted]     | 48/79 [03:34<02:18,  4.47s/it]

Number of matches 27908
Number of matches After Lowe's Ratio 1872
Number of Robust matches 676

 62%|[redacted]     | 49/79 [03:38<02:08,  4.29s/it]

Number of matches 27422
Number of matches After Lowe's Ratio 5297
Number of Robust matches 1893

 63%|[redacted]     | 50/79 [03:42<01:59,  4.13s/it]

Number of matches 26911
Number of matches After Lowe's Ratio 4800
Number of Robust matches 1491

 65%|[redacted]     | 51/79 [03:46<01:53,  4.04s/it]

Number of matches 25028
Number of matches After Lowe's Ratio 4345
Number of Robust matches 1673

 66%|[redacted]     | 52/79 [03:49<01:43,  3.83s/it]

Number of matches 25426
Number of matches After Lowe's Ratio 5848
Number of Robust matches 3050

 67%|[redacted]     | 53/79 [03:53<01:37,  3.75s/it]

Number of matches 26738
Number of matches After Lowe's Ratio 3453
Number of Robust matches 1976
```

```
68%|▉▉▉▉▉▉▉   | 54/79 [03:57<01:35,  3.83s/it]

Number of matches 27433
Number of matches After Lowe's Ratio 5992
Number of Robust matches 2559


70%|▉▉▉▉▉▉▉   | 55/79 [04:00<01:31,  3.81s/it]

Number of matches 27741
Number of matches After Lowe's Ratio 5824
Number of Robust matches 2612


Number of matches 28184
Number of matches After Lowe's Ratio 5385
71%|▉▉▉▉▉▉▉▉  | 56/79 [04:04<01:28,  3.87s/it]

Number of Robust matches 2923


72%|▉▉▉▉▉▉▉▉  | 57/79 [04:08<01:25,  3.90s/it]

Number of matches 29988
Number of matches After Lowe's Ratio 7204
Number of Robust matches 3762


73%|▉▉▉▉▉▉▉▉  | 58/79 [04:13<01:24,  4.04s/it]

Number of matches 31511
Number of matches After Lowe's Ratio 6552
Number of Robust matches 3840


75%|▉▉▉▉▉▉▉▉  | 59/79 [04:17<01:25,  4.25s/it]

Number of matches 32520
Number of matches After Lowe's Ratio 7793
Number of Robust matches 4495


76%|▉▉▉▉▉▉▉▉  | 60/79 [04:22<01:23,  4.40s/it]

Number of matches 31584
Number of matches After Lowe's Ratio 7426
Number of Robust matches 3508


77%|▉▉▉▉▉▉▉▉  | 61/79 [04:27<01:21,  4.50s/it]

Number of matches 31008
Number of matches After Lowe's Ratio 9486
Number of Robust matches 4415


78%|▉▉▉▉▉▉▉▉  | 62/79 [04:32<01:18,  4.59s/it]

Number of matches 32121
Number of matches After Lowe's Ratio 9387
Number of Robust matches 3907


80%|▉▉▉▉▉▉▉▉  | 63/79 [04:37<01:14,  4.65s/it]
```

```
Number of matches 33002
Number of matches After Lowe's Ratio 10025
Number of Robust matches 4301
```

```
 81%|[redacted] | 64/79 [04:42<01:11,  4.79s/it]
```

```
Number of matches 32555
Number of matches After Lowe's Ratio 10104
Number of Robust matches 3617
```

```
 82%|[redacted] | 65/79 [04:46<01:06,  4.74s/it]
```

```
Number of matches 29771
Number of matches After Lowe's Ratio 6387
Number of Robust matches 2009
```

```
 84%|[redacted] | 66/79 [04:51<01:00,  4.62s/it]
```

```
Number of matches 28928
Number of matches After Lowe's Ratio 7104
Number of Robust matches 2341
```

```
 85%|[redacted] | 67/79 [04:55<00:54,  4.57s/it]
```

```
Number of matches 31557
Number of matches After Lowe's Ratio 7497
Number of Robust matches 2405
```

```
 86%|[redacted] | 68/79 [05:00<00:51,  4.64s/it]
```

```
Number of matches 30819
Number of matches After Lowe's Ratio 9259
Number of Robust matches 3358
```

```
 87%|[redacted] | 69/79 [05:05<00:46,  4.65s/it]
```

```
Number of matches 31619
Number of matches After Lowe's Ratio 8356
Number of Robust matches 3088
```

```
 89%|[redacted] | 70/79 [05:09<00:41,  4.63s/it]
```

```
Number of matches 30391
Number of matches After Lowe's Ratio 5539
Number of Robust matches 2362
```

```
 90%|[redacted] | 71/79 [05:13<00:36,  4.55s/it]
```

```
Number of matches 30273
Number of matches After Lowe's Ratio 6508
Number of Robust matches 2903
```

```
 91%|[redacted] | 72/79 [05:18<00:31,  4.50s/it]
```

```
Number of matches 30462
Number of matches After Lowe's Ratio 6097
Number of Robust matches 2425
```

```
Number of Robust matches 2125
```

```
 92%|[redacted] | 73/79 [05:22<00:26,  4.42s/it]

Number of matches 30921
Number of matches After Lowe's Ratio 4980
Number of Robust matches 1790

 94%|[redacted] | 74/79 [05:27<00:22,  4.51s/it]

Number of matches 33337
Number of matches After Lowe's Ratio 4525
Number of Robust matches 1566

 95%|[redacted] | 75/79 [05:32<00:18,  4.66s/it]

Number of matches 34688
Number of matches After Lowe's Ratio 4789
Number of Robust matches 2023

 96%|[redacted] | 76/79 [05:37<00:14,  4.76s/it]

Number of matches 33208
Number of matches After Lowe's Ratio 421
Number of Robust matches 100

 97%|[redacted] | 77/79 [05:42<00:09,  4.83s/it]

Number of matches 31521
Number of matches After Lowe's Ratio 3455
Number of Robust matches 1360

 99%|[redacted] | 78/79 [05:46<00:04,  4.45s/it]

Number of matches 31633
Number of matches After Lowe's Ratio 7247
Number of Robust matches 3282
```

In [ ]:

```
H_left_agast = []
H_right_agast = []

num_matches_agast = []
num_good_matches_agast = []

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

    H_a,matches,gd_matches = get_Hmatrix(images_left_bgr[j:j+2][::-1],keypoints_all_left
_agast[j:j+2][::-1],points_all_left_agast[j:j+2][::-1],descriptors_all_left_agast[j:j+2]
[::-1])
    H_left_agast.append(H_a)
    num_matches_agast.append(matches)
    num_good_matches_agast.append(gd_matches)

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

```
    H_a,matches,gd_matches = get_Hmatrix(images_right_bgr[j:j+2][::-1],keypoints_all_rig
ht_agast[j:j+2][::-1],points_all_right_agast[j:j+2][::-1],descriptors_all_right_agast[j:
j+2][::-1])
    H_right_agast.append(H_a)
    num_matches_agast.append(matches)
    num_good_matches_agast.append(gd_matches)
```

In [ ]:

```
H_left_akaze = []
H_right_akaze = []

num_matches_akaze = []
num_good_matches_akaze = []

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

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

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

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

In [22]:

```
def warpnImages(images_left, images_right,H_left,H_right):
    #img1-centre,img2-left,img3-right

    h, w = images_left[0].shape[:2]

    pts_left = []
    pts_right = []

    pts_centre = np.float32([[0, 0], [0, h], [w, h], [w, 0]]).reshape(-1, 1, 2)

    for j in range(len(H_left)):
      pts = np.float32([[0, 0], [0, h], [w, h], [w, 0]]).reshape(-1, 1, 2)
      pts_left.append(pts)

    for j in range(len(H_right)):
      pts = np.float32([[0, 0], [0, h], [w, h], [w, 0]]).reshape(-1, 1, 2)
      pts_right.append(pts)

    pts_left_transformed=[]
    pts_right_transformed=[]

    for j,pts in enumerate(pts_left):
      if j==0:
        H_trans = H_left[j]
      else:
        H_trans = H_trans@H_left[j]
      pts_ = cv2.perspectiveTransform(pts, H_trans)
      pts_left_transformed.append(pts_)

    for j,pts in enumerate(pts_right):
      if j==0:
```

```
        H_trans = H_right[j]
      else:
        H_trans = H_trans@H_right[j]
      pts_ = cv2.perspectiveTransform(pts, H_trans)
      pts_right_transformed.append(pts_)


    print('Step1:Done')


    #pts = np.concatenate((pts1, pts2_), axis=0)

    pts_concat = np.concatenate((pts_centre,np.concatenate(np.array(pts_left_transformed
),axis=0),np.concatenate(np.array(pts_right_transformed),axis=0)), axis=0)

    [xmin, ymin] = np.int32(pts_concat.min(axis=0).ravel() - 0.5)
    [xmax, ymax] = np.int32(pts_concat.max(axis=0).ravel() + 0.5)
    t = [-xmin, -ymin]
    Ht = np.array([[1, 0, t[0]], [0, 1, t[1]], [0, 0, 1]])  # translate

    print('Step2:Done')


    return xmax,xmin,ymax,ymin,t,h,w,Ht
```

In [23]:

```
def final_steps_left_union(images_left,H_left,xmax,xmin,ymax,ymin,t,h,w,Ht):
    for j,H in enumerate(H_left):
        if j==  0:
            H_trans = Ht@H

        else:
            H_trans = H_trans@H
        result =  cv2.warpPerspective(images_left[j+1],H_trans,(xmax-xmin,ymax-ymin))
        warp_img_init_curr = result

        if j == 0:
            result[t[1]:h+t[1],t[0]:w+t[0]] = images_left[0]
            warp_img_init_prev = result
            continue
        black_pixels = np.where((warp_img_init_prev[:,:,0]==0)&(warp_img_init_prev[:,:,1
]==0)&(warp_img_init_prev[:,:,2]==0))
        warp_img_init_prev[black_pixels] = warp_img_init_curr[black_pixels]

    print('step31:Done')
    return warp_img_init_prev

def final_step_right_union(warp_img_prev,images_right,H_right,xmax,xmin,ymax,ymin,t,h,w,
Ht):
    for j,H in enumerate(H_right):
        if j==  0:
            H_trans = Ht@H
        else:
            H_trans = H_trans@H
        result =  cv2.warpPerspective(images_right[j+1],H_trans,(xmax-xmin,ymax-ymin))
        warp_img_init_curr = result



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

    print('step32:Done')
    return warp_img_prev
```

In [24]:

```
xmax,xmin,ymax,ymin,t,h,w,Ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_surfsift,H_right_surfsift)
```

```
Step1:Done
Step2:Done
```

In [25]:

```
warp_imgs_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_surfsift,xmax,x
min,ymax,ymin,t,h,w,Ht)
```

```
step31:Done
```

In [26]:

```
warp_imgs_all_surfsift = final_step_right_union(warp_imgs_left,images_right_bgr_no_enhanc
e,H_right_surfsift,xmax,xmin,ymax,ymin,t,h,w,Ht)
```

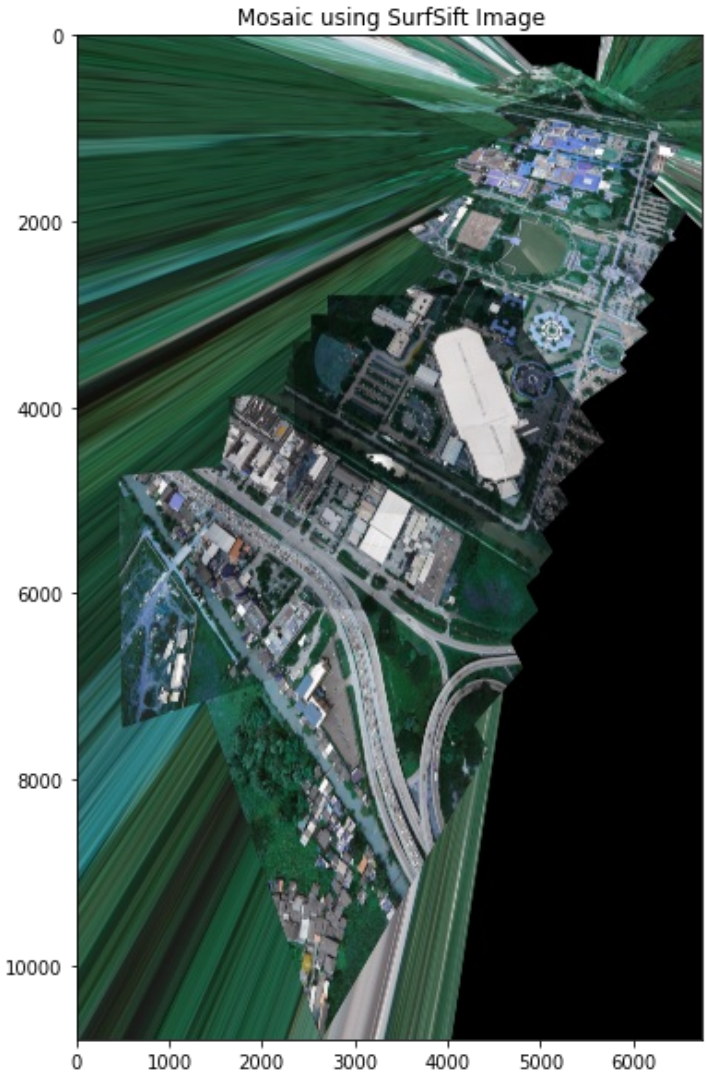
```
step32:Done
```

In [27]:

```
plt.figure(figsize=(20,10))
plt.imshow(warp_imgs_all_surfsift)
plt.title(' Mosaic using SurfSift Image')
```

Out[27]:

```
Text(0.5, 1.0, ' Mosaic using SurfSift Image')
```



In [ ]:

```
omax,omin,umax,umin,T,H,W,HT = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_gftt,H_right_gftt)
```

In [ ]:

```
warp_img = final_steps_left_union(images_left_bgr_no_enhance,H_left_gftt,omax,omin,umax,u
```

```
min,T,H,W,HT)
```

In [ ]:

```
warp_imgs_all_gftt = final_step_right_union(warp_img,images_right_bgr_no_enhance,H_right_
gftt,omax,omin,umax,umin,T,H,W,HT)
```

In [ ]:

```
plt.figure(figsize=(20,10))
plt.imshow(warp_imgs_all_gftt)
plt.title(' Mosaic using Gftt Image')
```

In [26]:

```
mmax,mmin,nmax,nmin,d,e,f,g = warpnImages(images_left_bgr_no_enhance, images_right_bgr_n
o_enhance,H_left_sift,H_right_sift)
```

```
Step1:Done
Step2:Done
```

In [27]:

```
warp_imgs_sift = final_steps_left_union(images_left_bgr_no_enhance,H_left_sift,mmax,mmin,
nmax,nmin,d,e,f,g)
```

```
step31:Done
```

In [28]:

```
warp_imgs_all_sift = final_step_right_union(warp_imgs_sift,images_right_bgr_no_enhance,H_
right_sift,mmax,mmin,nmax,nmin,d,e,f,g)
```

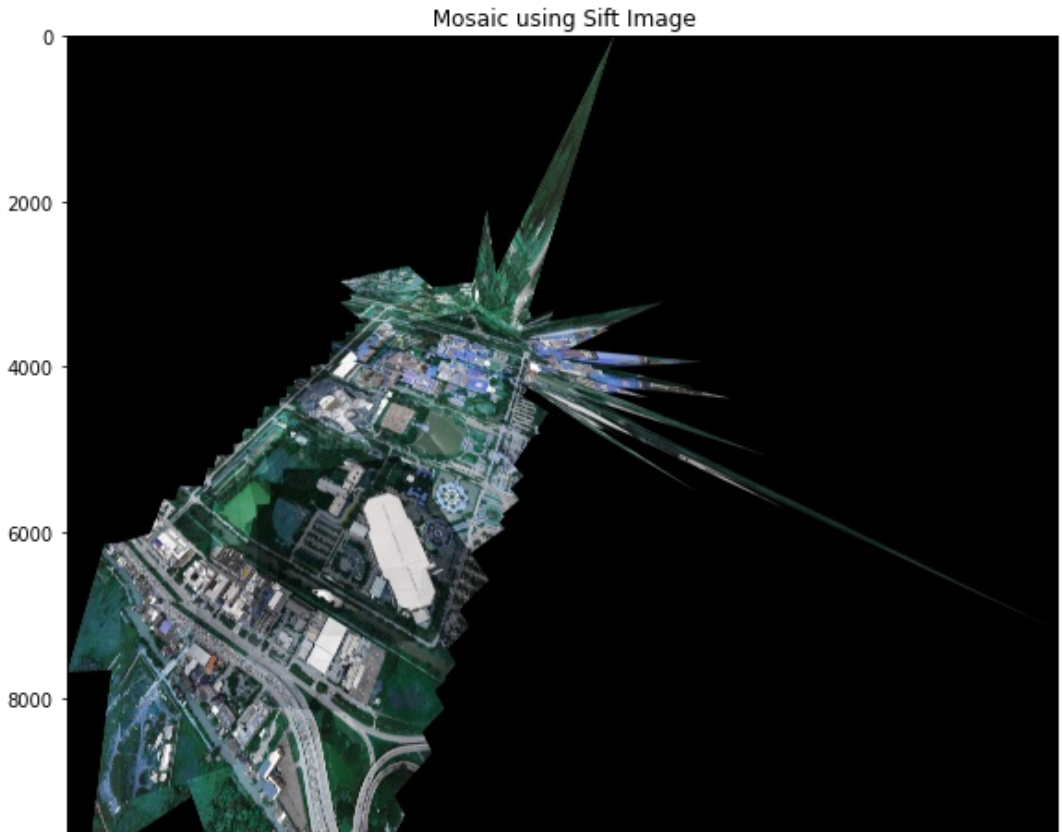
```
step32:Done
```

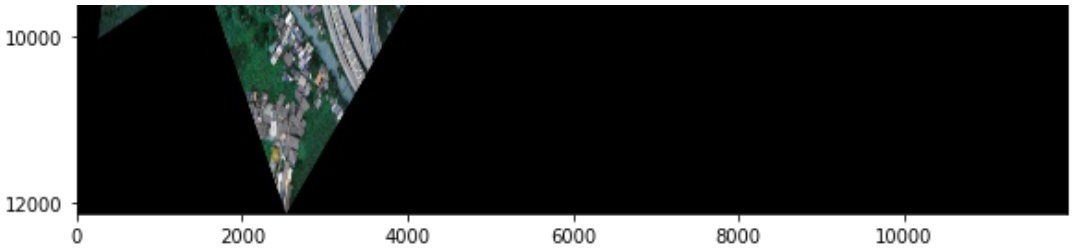
In [29]:

```
plt.figure(figsize=(20,10))
plt.imshow(warp_imgs_all_sift)
plt.title(' Mosaic using Sift Image')
```

Out[29]:

```
Text(0.5, 1.0, ' Mosaic using Sift Image')
```





In [23]:

```
omax,omin,umax,umin,T,H,W,HT = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_kaze,H_right_kaze)
```

```
Step1:Done
Step2:Done
```

In [24]:

```
warp_img_kaze = final_steps_left_union(images_left_bgr_no_enhance,H_left_kaze,omax,omin,u
max,umin,T,H,W,HT)
```

```
step31:Done
```

In [25]:

```
warp_imgs_all_kaze = final_step_right_union(warp_img_kaze,images_right_bgr_no_enhance,H_r
ight_kaze,omax,omin,umax,umin,T,H,W,HT)
```

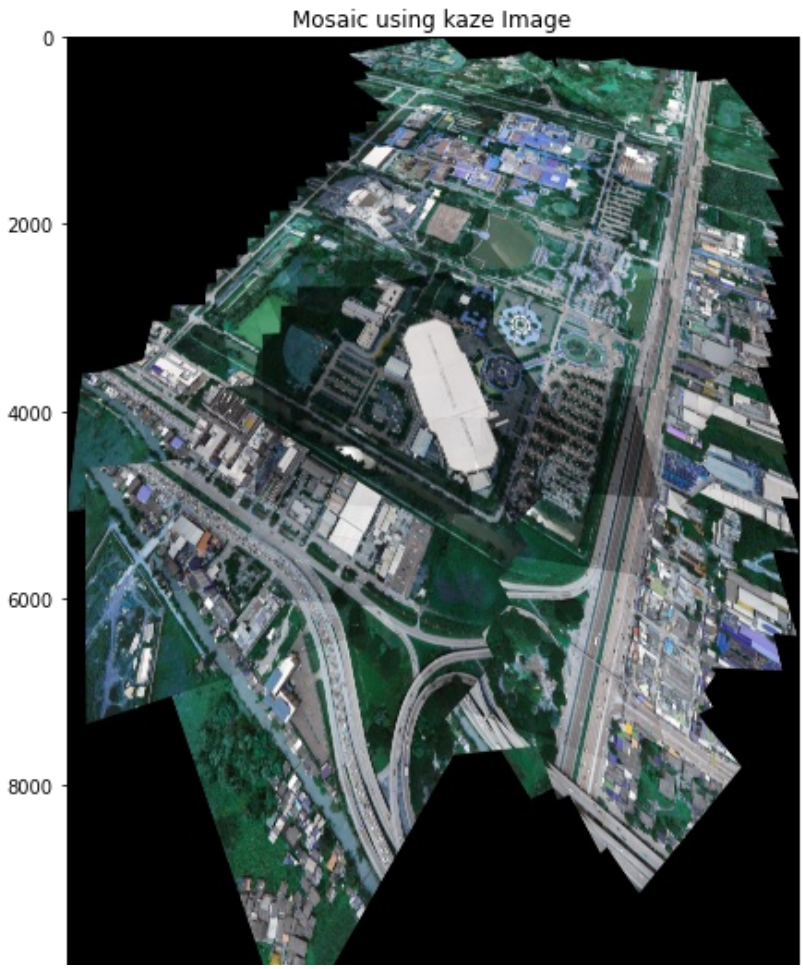
```
step32:Done
```

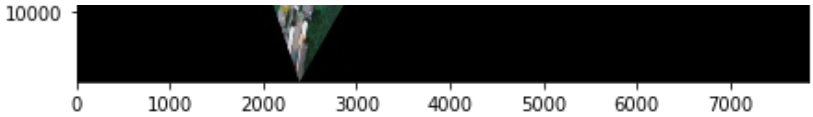
In [26]:

```
plt.figure(figsize=(20,10))
plt.imshow(warp_imgs_all_kaze)
plt.title('Mosaic using kaze Image')
```

Out[26]:

```
Text(0.5, 1.0, 'Mosaic using kaze Image')
```





In [ ]:

```
amax,amin,zmax,zmin,d,i,q,ht = warpnImages(images_left_bgr_no_enhance, images_right_bgr_
no_enhance,H_left_daisy,H_right_daisy)
```

In [ ]:

```
warp_image_left = final_steps_left_union(images_left_bgr_no_enhance,H_left_daisy,amax,ami
n,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
warp_imgs_all_daisy = final_step_right_union(warp_image_left,images_right_bgr_no_enhance,
H_right_daisy,amax,amin,zmax,zmin,d,i,q,ht)
```

In [ ]:

```
plt.figure(figsize=(20,10))
plt.imshow(warp_imgs_all_daisy)
plt.title('Mosaic using Daisy image')
plt.imsave('Mosaic using Daisy Image.jpg',warp_imgs_all_daisy)
```

In [ ]:

```

```