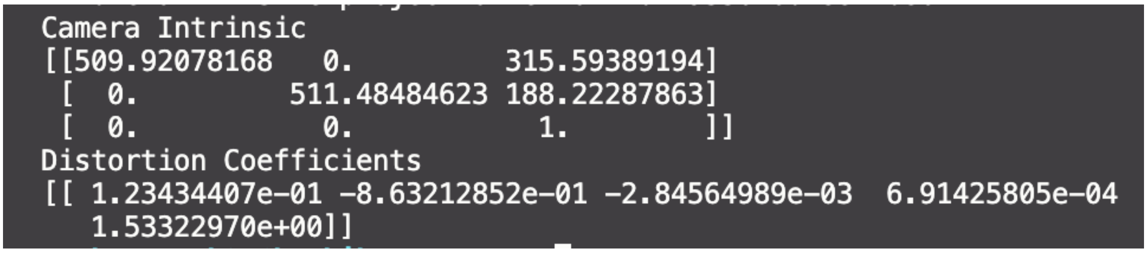
3dcv hw3 report

r09922115 朱世耘

Visualize result of each step:

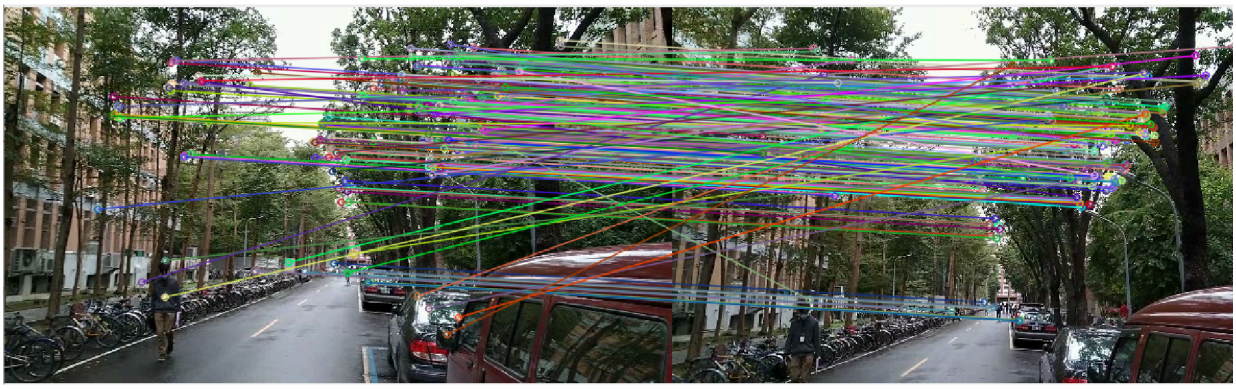
- camera calibration



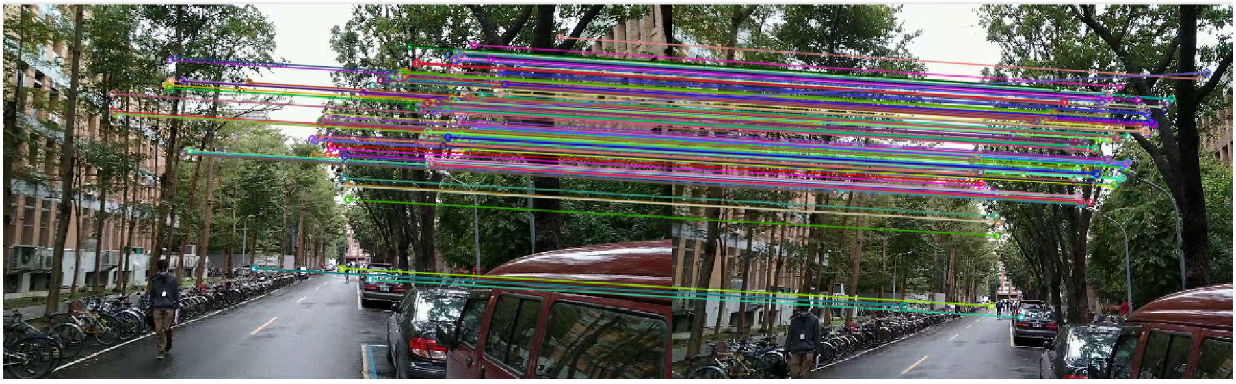
```
Camera Intrinsic
[[509.92078168   0.         315.59389194]
 [  0.         511.48484623 188.22287863]
 [  0.           0.           1.        ]]
Distortion Coefficients
[[ 1.23434407e-01 -8.63212852e-01 -2.84564989e-03  6.91425805e-04
   1.53322970e+00]]
```

- feature matching

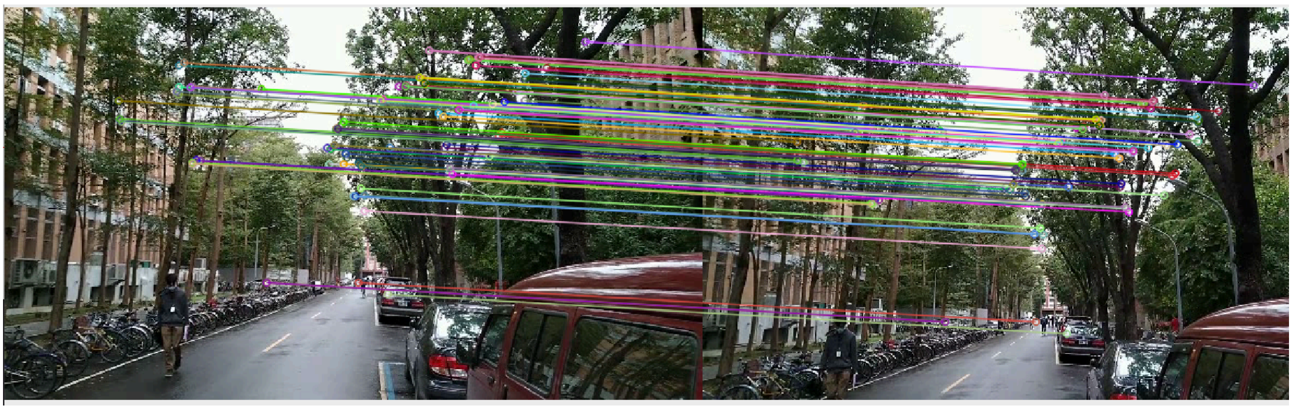
Corresponding key points from ORB



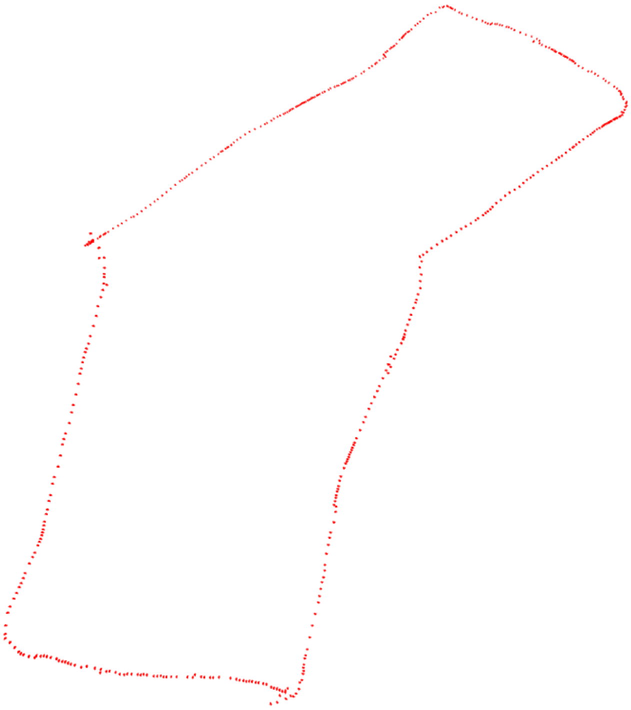
Inlier key points for finding essential matrix



Inlier key points for recover pose



Rescale t and plot camera pose



- Demo video

https://drive.google.com/file/d/1ipUg3B4kXd7G3VuwuHMfmW7wRzp_eFL6/view?usp=sharing

Discussion:

1. For feature matching, I use cv.ORB_create() and cv.BFMatcher(cv.NORM_HAMMING, crossCheck=True). From above image we can observe that it include come outliers.

2. Then, I use cv.findEssentialMat and cv.recoverPose. I sorted inlier in every step. We can see that there is almost no outlier.

3. I have to rescale t after cv.recoverPose, it is the hardest part in this work. I tried the method in the slide. At first, I randomly choose one scene points pair appear in both two images pair to calculate the scale of t. However, the t value change a lots when different pair are chosen. So I choose many pairs and calculate their median for final scale.

   $$\frac{\left\| {}^{k-1}\boldsymbol{t}_{k} \right\|}{\left\| {}^{k}\boldsymbol{t}_{k+1} \right\|} = \frac{\left\| {}^{k}\boldsymbol{X}_{k-1,k} - {}^{k}\boldsymbol{X}'_{k-1,k} \right\|}{\left\| {}^{k}\boldsymbol{X}_{k,k+1} - {}^{k}\boldsymbol{X}'_{k,k+1} \right\|}$$

4. To plot camera position and rotation, I use similar code as hw2.

To run my code, just type: ”python vo.py frames”