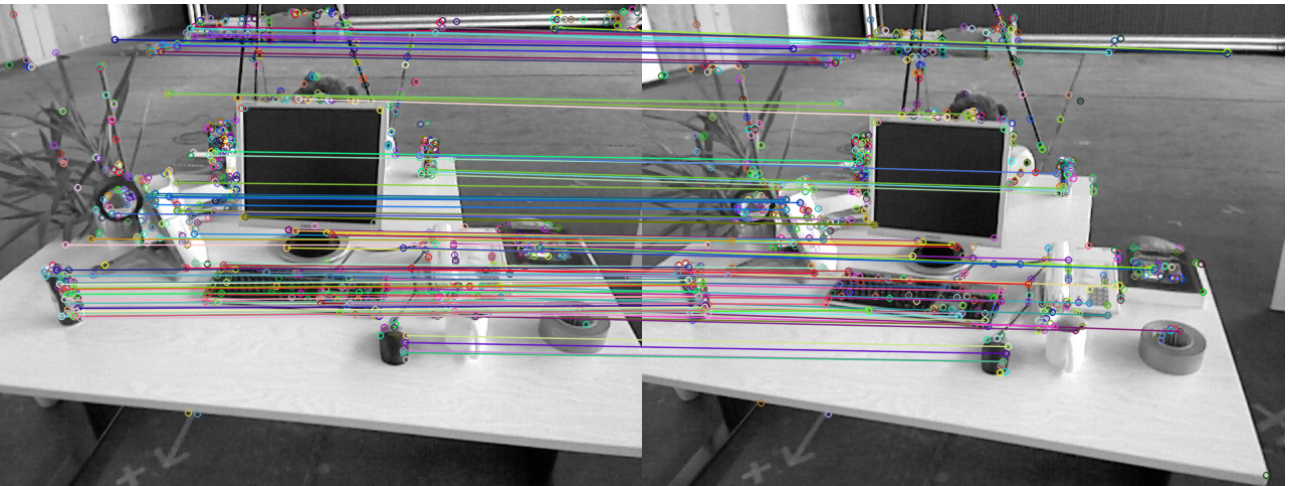


2.1-2.3 结果截图如下  
特征点如下



单线程匹配结果如下



2.4

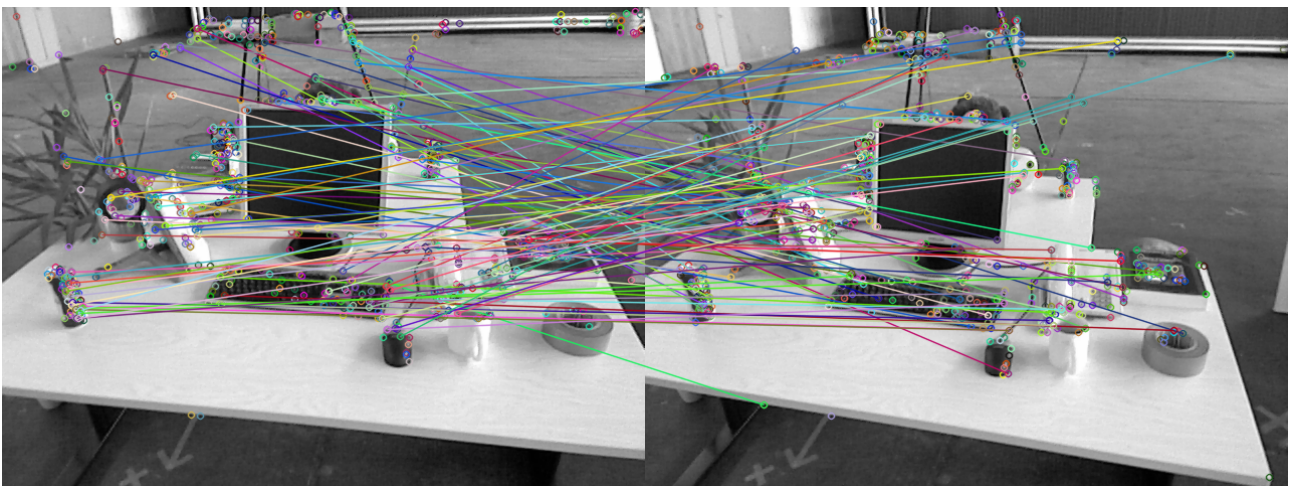
多线程运行时间如下

```
touchair@touchair-2020T:~/下载/视觉SLAM课程/L5/code/build$ ./computeORB
keypoints: 638
方法 compute angle mt_1 平均调用时间/次数: 3.94491/1 毫秒.
bad/total: 44/638
方法 compute orb descriptor mt_1 平均调用时间/次数: 3.22326/1 毫秒.
keypoints: 595
方法 compute angle mt_2 平均调用时间/次数: 2.07726/1 毫秒.
bad/total: 7/595
方法 compute orb descriptor mt_2 平均调用时间/次数: 7.78381/1 毫秒.
方法 bf match 平均调用时间/次数: 2412.2/1 毫秒.
matches: 95
```

单线程运行时间如下

```
touchair@touchair-2020T:~/下载/视觉SLAM课程/L5/code/build$ ./computeORB
keypoints: 638
方法 compute angle_1 平均调用时间/次数: 3.71051/1 毫秒.
bad/total: 44/638
方法 compute orb descriptor_1 平均调用时间/次数: 19.0201/1 毫秒.
keypoints: 595
方法 compute angle_2 平均调用时间/次数: 3.50776/1 毫秒.
bad/total: 7/595
方法 compute orb descriptor_2 平均调用时间/次数: 18.7916/1 毫秒.
方法 bf match 平均调用时间/次数: 2301.58/1 毫秒.
matches: 95
```

多线程匹配结果如下



#### 2.4.1

其描述子是以二进制表示的

#### 2.4.2

阈值取得过大会导致误匹配率增加；阈值取得过小会导致匹配对过少。这两种情况会导致后续的位姿求解不稳定。

#### 2.4.3

需要2秒左右，后续可以采用快速最近邻匹配法。

#### 2.4.4

多线程和单线程计算旋转角耗时差不多，计算描述子多线程耗时更短

3 运行结果如下图所示

```
touchair@touchair-2020T:~/下载/视觉SLAM课程/L5/code/build$ ./E2Rt
R1 =
-0.998596  0.0516992 -0.0115267
-0.0513961 -0.99836 -0.0252005
 0.0128107  0.0245727 -0.999616
R2 =
-0.365887 -0.0584576  0.928822
-0.00287462  0.998092  0.0616848
 0.930655 -0.0198996  0.365356
t1 =
-0.581301
-0.0231206
 0.401938
t2 =
 0.581301
 0.0231206
-0.401938
t^R =
 0.0203619  0.400711  0.0332407
-0.393927  0.035064 -0.585711
 0.00678849  0.581543  0.0143826
touchair@touchair-2020T:~/下载/视觉SLAM课程/L5/code/build$
```

4

```
touchair@touchair-2020T:~/下载/视觉SLAM课程/L5/code/build$ ./GN-BA
points: 76
iteration 0 cost=622769.1141257
iteration 1 cost=12206.604278533
iteration 2 cost=12150.675954556
iteration 3 cost=12150.675326912
iteration 4 cost=12150.67532691
cost: 150.67532691, last cost: 150.67532691
estimated pose:
 0.997866187875 -0.0516724155459  0.0399128120605 -0.12722662332
 0.0505958883589  0.998339767436  0.0275275287143 -0.00750705825792
-0.0412689614128 -0.0254493659567  0.998823909704  0.0613860392383
      0          0          0          1
```

4.1 第*i*个投影点的误差为

$$e_i = u_i - \frac{1}{s_i} K \exp(\xi^\wedge) P_i$$

重投影误差如下

$$\begin{aligned} e &= \frac{1}{2} \sum_{i=1}^n \|e_i\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n \left\| u_i - \frac{1}{s_i} K \exp(\xi^\wedge) P_i \right\|^2 \end{aligned}$$

4.2 雅克比矩阵计算如下

$$\begin{aligned}\frac{\partial e}{\partial \delta \xi} &= \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial \delta \xi} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} [I, -P'^\wedge] \\ &= - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & Z' & -Y' \\ 0 & 1 & 0 & -Z' & 0 & X' \\ 0 & 0 & 1 & Y' & -X' & 0 \end{bmatrix} \\ H = \frac{\partial e}{\partial \xi} &= - \begin{bmatrix} \frac{f_x}{Z'} & 0 & \frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & \frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y X' Y'}{Z'^2} & \frac{f_y X^2}{Z'^2} & -\frac{f_y Y'}{Z'} \end{bmatrix}\end{aligned}$$

4.3 更新如下

```
T_esti = Sophus::SE3d::exp(dx) * T_esti;
```

5 对准前后 RMSE 误差

```
touchair@touchair-2020T:~/下载/视觉SLAM课程/L5/code/build$ ./drawTrajectory
rmse: 3.72068
rmse: 0.0415122
```

对准前后轨迹

