视觉 SLAM 理论与实践 - 作业 5

peng00bo00

June 18, 2020

1. 代码可参见./code/computeORB.cpp
   提取到的特征点和匹配结果如 Fig.1，Fig.2所示

Figure 1: ORB 特征点

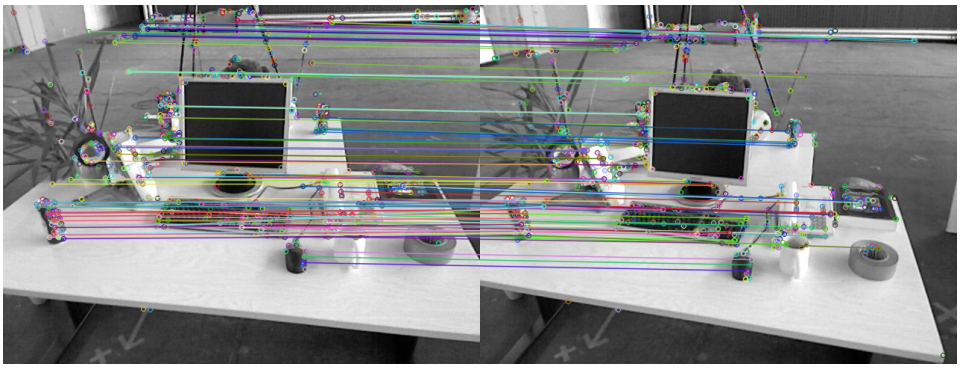

Figure 2: 特征匹配

1) ORB 特征的描述子为一个 256 维的 bool 型向量，可以用二进制来进行代替，因此是一种二进制特征

2) 阈值表示对特征点描述子发生变化的敏感程度，增加阈值意味着放宽对特征点描述子相似性的要求，进而会得到更多的匹配；反之降低阈值相当与要求描述子之间具有更大的相似性，因此会过滤掉一些匹配结果从而减少匹配数

3) 要提高匹配速度可以使用更先进的匹配算法和数据结构，如 FLANN 使用 KD 树来加速匹配 i 过程；或者在匹配时进行并行，每个特征点独立进行匹配。

2. 代码可参见./code/E2Rt.cpp
   对应的旋转矩阵和平移向量为

$$R_1 = \begin{bmatrix} -0.365887 & -0.0584576 & 0.928822 \\ -0.00287462 & 0.998092 & 0.0616848 \\ 0.930655 & -0.0198996 & 0.365356 \end{bmatrix} \tag{1}$$

$$R_2 = \begin{bmatrix} -0.998596 & 0.0516992 & -0.0115267 \\ -0.0513961 & -0.99836 & -0.0252005 \\ 0.0128107 & 0.0245727 & -0.999616 \end{bmatrix} \tag{2}$$

$$t_1 = \begin{bmatrix} -0.581301 \\ -0.0231206 \\ 0.401938 \end{bmatrix} \tag{3}$$

$$t_2 = \begin{bmatrix} 0.581301 \\ 0.0231206 \\ -0.401938 \end{bmatrix} \tag{4}$$

3. 代码可参见./code/GN-BA.cpp

1) 重投影误差是指三维空间中的点经过相机投影到图像上后实际投影点与通过相机内外参矩阵计算得到的投影点之间的位置误差，如果相机的内参和外参矩阵完全正确，理论上重投影误差应该为 0。由于计算投影点位置时需要用到相机的位姿 (外参矩阵) 因此可以通过优化重投影误差来计算相机的位姿。重投影误差的计算式为:

$$L = \frac{1}{2}\sum_{i=1}^{n} ||u_i - \frac{1}{s_i}KTP_i||^2 \tag{5}$$

其中，u_i 为三维点在二维图像上的坐标；K 为相机内参矩阵；T 为相机的位姿；P_i 为三维点在空间中的坐标。

2) 使用扰动模型得到雅可比矩阵为

$$J = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix} \tag{6}$$

其中 (X', Y', Z') 为三维空间点在相机坐标系下的坐标。

3) 解出更新量后可根据扰动模型更新相机位姿

$$T \leftarrow \exp\{\xi^\wedge\}T \tag{7}$$

最终得到相机的位姿为

$$R_2 = \begin{bmatrix} 0.997866186837 & -0.0516724392948 & 0.0399128072707 & -0.127226620999 \\ 0.0505959188721 & 0.998339770315 & 0.0275273682287 & -0.00750679765283 \\ -0.041268949107 & -0.0254492048094 & 0.998823914318 & 0.0613860848809 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

4. 代码可参见./code/ICP.cpp
   轨迹对齐结果如 Fig.3所示

Figure 3: 轨迹对齐