

视觉 SLAM 理论与实践 - 作业 7

peng00bo00

July 24, 2020

1. (a)
 - i. 由于 BA 问题具有稀疏性, 在求解 BA 问题时会比想象中要高效得多
 - ii. BA 中需要进行参数化的内容包括 3D 特征, 3D 旋转, 相机标定以及一些非线性约束等。
3 维点可以使用齐次坐标或非齐次坐标来进行参数化。非齐次坐标处理起来较为简单, 但对于比较远的点可能会带来较大的误差, 此时使用齐次坐标会更为方便。
位姿 (旋转) 可以使用单位四元数或是扰动模型来进行参数化, 需要注意的是使用单位四元数时要考虑额外的约束
 - iii. BA 问题的网络结构 (Network Structure) 为后来图优化的诞生奠定了基础, 同时对 BA 问题稀疏性的研究也使得高效求解 BA 问题变得可能
- (b) 代码可参见./code/bundle_adjustment_g2o.cpp 这里在书上给出代码的基础上添加了计算雅可比矩阵的部分, 最后得到点云如 Fig.1所示

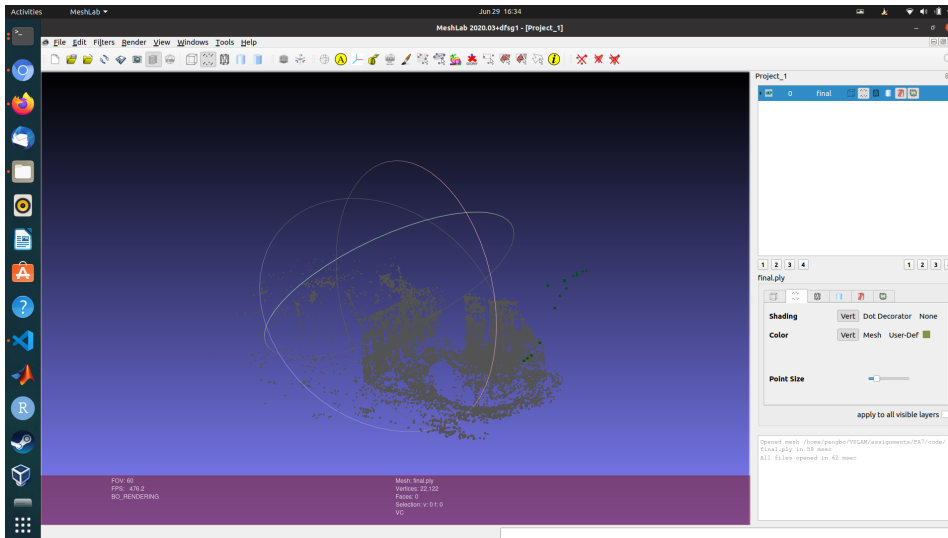


Figure 1: 点云

- (c) 补充: 代码中使用了理论导数进行计算, 当结果收敛后会出现 Cholesky failed 的错误, 如果使用数值导数则不会报错。在新一版的代码中减少了迭代计算的次数来避免 Cholesky failed.

2. 代码可参见./code/directBA.cpp, 最终得到点云如 Fig.2所示

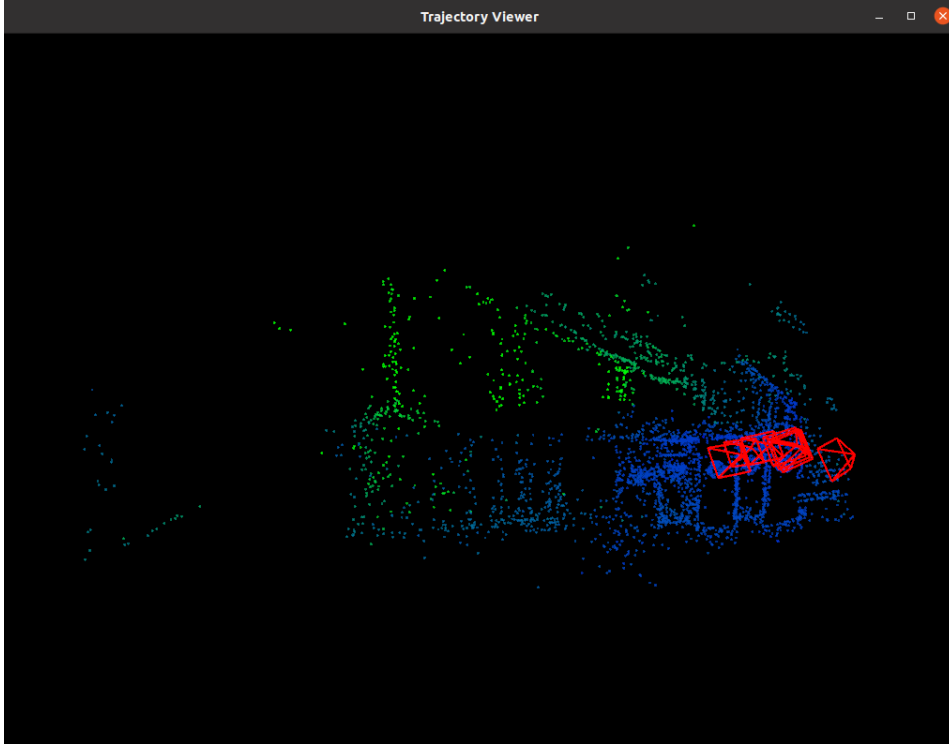


Figure 2: 直接法点云

- (a) i. 假设空间点 $P = (x, y, z)$ 对应的灰度为 I , 其投影到图像上的坐标为 $p = (u, v)$ 对应灰度为 $I'(u, v)$, 则该点对应的误差项为
- ii. 误差 e 与该点的空间坐标 (x, y, z) 以及相机的位姿 T 有关
- iii. 设点 $P = (x, y, z)$ 在相机坐标系下的坐标为 q , 投影到图像上的坐标为 u , 相机的内参矩阵为 K , 有

$$e = I - I'(u, v) \quad (1)$$

$$q = TP \quad (2)$$

$$u = \frac{1}{q_z} Kq \quad (3)$$

因此对应的雅可比矩阵为

$$J_\xi = -\frac{\partial I'}{\partial u} \frac{\partial u}{\partial q} \frac{\partial q}{\partial \xi} \quad (4)$$

$$J_P = -\frac{\partial I'}{\partial u} \frac{\partial u}{\partial q} \frac{\partial q}{\partial P} \quad (5)$$

其中 J_ξ 为误差项对相机位姿李代数的雅可比矩阵而 J_P 为误差项对空间点 P 的雅可比矩阵. J_ξ 与书上直接法推导完全相同而 J_P 仅在最后一步 $\frac{\partial q}{\partial P}$ 与 J_ξ 不同. 假设位姿 T 对应的旋转矩阵为 R , 则

$$\frac{\partial q}{\partial P} = R \quad (6)$$

只需用 R 取代 $\frac{\partial q}{\partial \xi}$ 即可.

- (b) i. 直接法对如何参数化点没有进行限制, 只要能够计算误差对空间点的导数即可
- ii. 4×4 的结果还可以, 理论上讲 patch 取的越大平均误差会越小
- iii. 特征点法是最小化重投影误差, 而直接法是最小化光度误差
- iv. 本题中 Huber 函数的阈值取了 0.5 用于降低外点对误差的贡献
- (c) 补充: 新一版的代码中修改了添加边的策略, 使得每一个相机位姿都对应全部的空间点. 在计算损失时则仅考虑空间点及其周边能够投影到图像内的部分, 若投影后超出图像边界则该部分损失为 0.
- (d) 补充: 经检查代码在添加空间点颜色时有错误, 修正后得到点云如 Fig.3所示

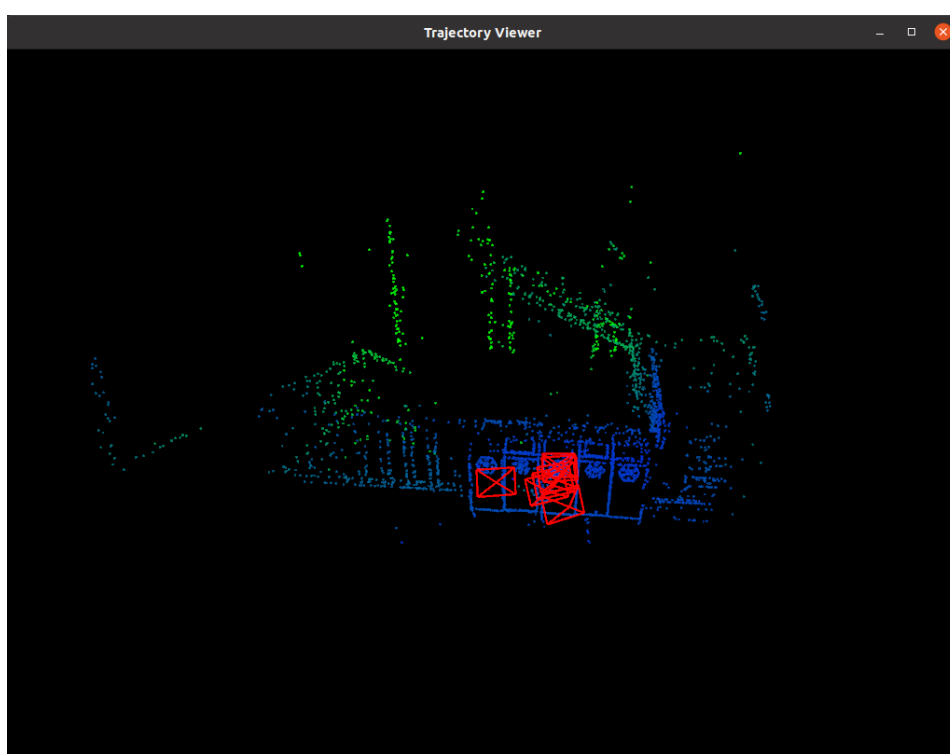


Figure 3: 直接法点云