

Task3

2-0、配置与说明

从 github 网站 <https://github.com/weisui-ad/ImageBasedModellingEdu.git> 上拉取最新的代码, 注意仔细处理冲突, 不要覆盖自己本地的已有代码。

2-1 实现线性三角化算法(Linear triangulation methods)

给定匹配点以及相机投影矩阵(至少 2 对), 计算对应的三维点坐标。当给定相机内外参矩阵时, 图像上每个特征点实际上对应三维中一条射线, 理想情况下, 利用两条射线相交便可以得到三维点的坐标。但是实际中, 由于计算或者检测误差, 无法保证两条射线的相交性, 因此需要建立新的数学模型(如最小二乘)进行求解。

考虑两个视角的情况, 假设空间中的三维点P的齐次坐标为 $\mathbf{X} = [x, y, z, 1]^T$, 对应地, 在两个视角的投影点分别为 p_1 和 p_2 , 它们的图像坐标为

$$\mathbf{x}_1 = [x_1, y_1, 1]^T, \mathbf{x}_2 = [x_2, y_2, 1]^T$$

两幅图像对应的相机投影矩阵为 P_1, P_2 (P_1, P_2 维度是 3×4), 理想情况下

$$\mathbf{x}_1 = P_1 \mathbf{X}, \mathbf{x}_2 = P_2 \mathbf{X}$$

考虑第一个等式, 在其两侧分别叉乘 \mathbf{x}_1 , 可以得到

$$\mathbf{x}_1 \times (P_1 \mathbf{X}) = \mathbf{0},$$

将 $P_1 \mathbf{X}$ 表示成 $[P_{11} \mathbf{X}, P_{21} \mathbf{X}, P_{31} \mathbf{X}]^T$, 其中 P_{11}, P_{21}, P_{31} 分别是投影矩阵 P_1 的第 1-3 行, 我们可以得到

$$x_1(P_{13} \mathbf{X}) - P_{11} \mathbf{X} = 0$$

$$y_1(P_{13} \mathbf{X}) - P_{12} \mathbf{X} = 0$$

$$x_1(P_{12} \mathbf{X}) - y_1(P_{11} \mathbf{X}) = 0$$

其中第三个方程可以由前两个通过线性变换得到, 因此我们只考虑前两个方程。每一个视角可以提供两个约束, 联合第二个视角的约束, 我们可以得到

$$A \mathbf{X} = \mathbf{0}$$

其中

$$A = \begin{bmatrix} x_1 P_{13} - P_{11} \\ y_1 P_{13} - P_{12} \\ x_2 P_{23} - P_{21} \\ y_2 P_{23} - P_{22} \end{bmatrix}$$

当视角个数多于 2 个的时候, 可以采用最小二乘的方式进行求解, 理论上, 在不存在外点的情况下, 视角越多估计的三维点坐标越准确。当存在外点(错误

的匹配点)时, 则通常采用 RANSAC 的鲁棒估计方法进行求解。

参考上述原理, 实现 task3/class3_test_triangle.cc 中 A 矩阵的构造。打印并比对结果。

2-1 推导并实现 Jacobian 矩阵(重点)

参考附件, 《BA Jacobian 矩阵的推导》推导 Jacobian 矩阵, 并完成代码 task3/class3_test_jacobian.cc 中求 Jacobian 矩阵的函数, 打印并比对。

```
void jacobian(sfm::ba::Camera const& cam, sfm::ba::Point3D const& point, double* cam_x_ptr, double* cam_y_ptr, double*
point_x_ptr, double* point_y_ptr);
```

2-2 推导并实现 P3p 算法以及基于 RANSAC 鲁棒算法(重点)

运行代码 task2-2_test_p3p_kneip.cc 和 task2-2_test_p3p_ransac.cc, 了解并掌握 p3p 算法的原理。

2-3 熟悉并掌握 Levenberg-Marquardt。

参考 slides 中 LM 算法流程, 运行 task2/task2-3_test_lm_optimize.cc 中的函数

```
void lm_optimization(std::vector<sfm::ba::Camera>* cameras
std::vector<sfm::ba::Point3D>* points, std::vector<sfm::ba::Observation>* observations)
```

打印输出结果并与正确结果进行比对, 并自行写出算法伪代码。

2-4 熟悉并掌握 BA 的算法原理, 包括雅可比计算过程。

参考《BA 雅可比矩阵推导.pdf》, 实现代码 task2/task2-4_test_jacobian.cc 函数

2-5 一个完整的双视角 SFM 过程

完成了特征点提取与匹配, 相机基础矩阵的求取与相机姿态的恢复, 三维点的三角量测, 以及相机姿态与三维点坐标的非线性优化(捆绑调整/集束调整)。其中焦距信息目前是从图像 Exif 头信息文件中读取。调试 task2/task2-5_test_bundle_adjustment.cc 工程, 观察输出结果。