

## Task3: 双视角SFM

### 配置与说明

从 github 网站 <https://github.com/swayfreeda/ImageBasedModellingEduV1.0.git>

上下载工程文件, 里面包含本次课的作业和代码范例(examples/task3)文件夹下。Clion 软件可以配置 git 版本控制。

将上述工程 fork 到自己的 github 账户下

首先需要安装 git 软件

然后从 Clion 菜单 VCS->Git->Clone 进行拷贝

### 3-1 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  $X = [x, y, z, 1]^T$ , 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  $3 \times 4$ ), 理想情况下

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

考虑第一个等式, 在其两侧分别叉乘  $\mathbf{x}_1$ , 可以得到

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

将  $P_1 X$  表示成  $[P_{11}X, P_{21}X, P_{31}X]^T$ , 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

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

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

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

其中第三个方程可以由前两个通过线性变换得到, 因此我们只考虑全两个方程。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

$$AX = \mathbf{0}$$

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。

### 3-2 推导并实现雅阁比矩阵

参考附件, 《BA 雅阁比矩阵的推导》推导雅阁比矩阵, 并完成代码 task3/class3\_test\_jacobian.cc 中求雅阁比矩阵的函数, 打印并比对结果。

```
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);
```

### 3-3 熟悉并掌握 Levenberg-Marquardt。

参考 slides 中 LM 算法流程, 运行 task3/class3\_test\_jacobian.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)
```

打印输出结果并余正确结果进行比对。并自行写出算法伪代码。

### 3-4 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 task3/class3\_test\_bundle\_adjustment.cc 工程, 观察输出结果。