第三章作业思路分享

主讲人 刘哲铭

深蓝学院 shenlanxueyuan.com

1. 雅克比推导：推导点到直线距离关于T的雅克比

目标函数：

$$d_e = \frac{|(\tilde{p}_i - p_a) \times (\tilde{p}_i - p_b)|}{|p_a - p_b|}$$

其中, $\tilde{p}_i = Tp = exp(\xi^\wedge)p$

令 $f(T) = (\tilde{p}_i - p_a) \times (\tilde{p}_i - p_b)$

$$===> \frac{\partial d_e}{\partial T} = \frac{\partial d_e}{\partial |f|} * \frac{\partial |f|}{\partial f} * \frac{\partial f}{\partial \tilde{p}_i} * \frac{\partial \tilde{p}_i}{\partial T}$$

其中：

$$\frac{\partial d_e}{\partial |f|} = \frac{1}{|p_a - p_b|}$$

$$\frac{\partial |f|}{\partial f} = \frac{\partial sqrt(f^T f)}{\partial f} = \frac{\partial sqrt(f^T f)}{\partial (f^T f)} * \frac{\partial (f^T f)}{\partial (f)} = \frac{f}{|f|}$$

$$\frac{\partial f}{\partial \tilde{p}_i} = \frac{\partial (\tilde{p}_i - p_a)}{\partial \tilde{p}_i} \times (\tilde{p}_i - p_b) + (\tilde{p}_i - p_a) \times \frac{\partial (\tilde{p}_i - p_b)}{\partial \tilde{p}_i} = (p_b - p_a)^\wedge$$

$$\frac{\partial \tilde{p}_i}{\partial T} = \frac{\partial (Tp)}{\partial \delta\xi} = \lim_{\delta\xi \to 0} \frac{exp(\delta\xi^\wedge)exp(\xi^\wedge)p - exp(\xi^\wedge)p}{\delta\xi}$$

$$\approx \lim_{\delta\xi \to 0} \frac{(I + \delta\xi^\wedge)exp(\xi^\wedge)p - exp(\xi^\wedge)p}{\delta\xi}$$

$$= \lim_{\delta\xi \to 0} \frac{\delta\xi^\wedge exp(\xi^\wedge)p}{\delta\xi}$$

$$= \begin{bmatrix} I & -(Rp + t)^\wedge \\ 0^T & 0 \end{bmatrix}$$

深蓝学院 shenlanxueyuan.com

ℓ **1. 雅克比推导：**推导点到平面距离关于T的雅克比

目标函数：

$$d_h = (\tilde{p}_i - p_j) * \frac{(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)}{|(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)|}$$

其中，$\frac{(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)}{|(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)|}$表示平面的单位法向量

$$\frac{\partial d_h}{\partial T} = \frac{\partial d_h}{\partial \tilde{p}_i} * \frac{\partial \tilde{p}_i}{\partial T}$$

$$= \frac{(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)}{|(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)|} * \frac{\partial \tilde{p}_i}{\partial T}$$

$$= \frac{(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)}{|(\tilde{p}_l - p_j) \times (\tilde{p}_m - p_j)|} * [I \quad -(Rp + t)^\wedge]$$

2. ceres使用

不少同学都是采用Floam的形式，使用1*7的参数传递给ceres，但是也可以用1,4,3的形式进行传递，就是残差维度是1维，四元数数据输入为4维，然后位置输入为3维。

基于Floam那种方式的只需要将对应的代码放入到现在的框架中即可，这里就不做过多讲解，而基于四元数和平移量两个参数块的形式的，主要问题在于jacobian怎么去写

2. ceres使用

不少同学都是采用Floam的形式，使用1*7的参数传递给ceres，但是也可以用1,4,3的形式进行传递，就是残差维度是1维，四元数数据输入为4维，然后位置输入为3维。

基于Floam那种方式的只需要将对应的代码放入到现在的框架中即可，这里就不做过多讲解，而基于四元数和平移量两个参数块的形式的，主要问题在于jacobian怎么去写

注意点1：ceres::EigenQuaternionParameterization参数，全局维度是4维，但是实际参与求导运算的是3维

注意点2：ceres中，jacobian**是二重指针，有几个参数块，就对应着有几个雅克比相对应

2. ceres使用

```
class PointToLineCost : public ceres::SizedCostFunction<1, 4, 3> {
```

举例说明：

右图中为点到面的距离残差

残差与输入参数块的维度为3,4,3

在重载Evaluate函数的时候，

1.根据前面公式构建残差

2. 更新jacobians[0]，即对旋转的雅克比

3. 更新jacobians[1]，即对平移的雅克比

```
bool PointToPlaneCost::Evaluate(double const *const *parameters,
                                double *residuals, double **jacobians) const
                                {

  // @TODO: construct residuals
  residuals[0];

  if (jacobians != NULL) {
    double *jacobian = jacobians[0]; // jacobian ref to first parameter_block
    if (jacobian != NULL) {
      jacobian[0] = // dqx;
      jacobian[1] = // dqy;
      jacobian[2] = // dqz;
      jacobian[3] = 0; // dqw;
    }
    jacobian = jacobians[1]; // jacobian ref to second parameter_block
    if (jacobian != NULL) {
      jacobian[0] = // dx;
      jacobian[1] = // dy;
      jacobian[2] = // dz;
    }
  }
  return true;
}
```

3. 轨迹对比

这个只需要将跑完的gps轨迹和odom轨迹使用EVO工具对比一下就可以了，结合第二章的课件，就可以实现了