0. 说明

本 PDF 文档为自动生成，如有遗漏的格式错误但不影响阅读请见谅，若影响了阅读请告知！

问1：设目标函数为

$$J(\boldsymbol{q},\boldsymbol{r},\lambda)=\frac{1}{2}\sum_{j=1}^{M} w_j \boldsymbol{q}^T(\boldsymbol{y}_j^{\oplus}-(\boldsymbol{p}_j-\boldsymbol{r})^{+})^T(\boldsymbol{y}_j^{\oplus}-(\boldsymbol{p}_j-\boldsymbol{r})^{+})\boldsymbol{q}-\frac{1}{2}\lambda(\boldsymbol{q}^T\boldsymbol{q}-1)$$

请证明令目标函数对 $\boldsymbol{r}$ 的偏导为0可得到 $\boldsymbol{r}=\boldsymbol{p}-\boldsymbol{q}^{+}\boldsymbol{y}^{+}\boldsymbol{q}^{-1}$

其中 $\boldsymbol{y}=\frac{1}{w}\sum_{j=1}^{M} w_j\boldsymbol{y}_j$， $\boldsymbol{p}=\frac{1}{w}\sum_{j=1}^{M} w_j\boldsymbol{p}_j$， $w=\sum_{j=1}^{M} w_j$

证： 由题目可得：

$$\frac{\partial J}{\partial \boldsymbol{r}^T}=\boldsymbol{q}^{-1\oplus}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j^{\oplus}-(\boldsymbol{p}_j-\boldsymbol{r})^{+})\boldsymbol{q}$$

$$=\sum_{j=1}^{M} w_j\boldsymbol{q}^{-1\oplus}\boldsymbol{y}_j^{\oplus}\boldsymbol{q}-\sum_{j=1}^{M} w_j\boldsymbol{q}^{-1\oplus}\boldsymbol{p}_j^{+}\boldsymbol{q}+\sum_{j=1}^{M} w_j\boldsymbol{q}^{-1\oplus}\boldsymbol{r}^{+}\boldsymbol{q} \qquad \text{(1-1)}$$

$\boldsymbol{q}^{-1\oplus}\boldsymbol{y}_j^{\oplus}\boldsymbol{q}=(\boldsymbol{q}^{-1\oplus}\boldsymbol{y_j})^{\oplus}\boldsymbol{q}=\boldsymbol{q}^{+}(\boldsymbol{q}^{-1\oplus}\boldsymbol{y_j})=\boldsymbol{q}^{+}\boldsymbol{y_j}^{+}\boldsymbol{q}^{-1}$ (1-2)

$\boldsymbol{q}^{-1\oplus}\boldsymbol{p}_j^{+}\boldsymbol{q}=\boldsymbol{q}^{-1\oplus}\boldsymbol{q}^{\oplus}\boldsymbol{p_j}=\boldsymbol{1}\boldsymbol{p_j}=\boldsymbol{p_j}$ (1-3)

$\boldsymbol{q}^{-1\oplus}\boldsymbol{r}^{+}\boldsymbol{q}=\boldsymbol{q}^{-1\oplus}\boldsymbol{q}^{\oplus}\boldsymbol{r}=\boldsymbol{1}\boldsymbol{r}=\boldsymbol{r}$ (1-4)

将公式（1-2、1-3、1-4）带入（1-1）可得：

$\sum_{j=1}^{M} w_j\boldsymbol{q}^{+}\boldsymbol{y_j}^{+}\boldsymbol{q}^{-1}-\sum_{j=1}^{M} w_j\boldsymbol{p_j}+\sum_{j=1}^{M} w_j\boldsymbol{r}$ (1-5)

令（1-5）等于零可得：$\boldsymbol{r}=\boldsymbol{p}-\boldsymbol{q}^{+}\boldsymbol{y}^{+}\boldsymbol{q}^{-1}$

问2：请证明 $\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z_j}^{\odot^T}\boldsymbol{z_j}^{\odot} = \mathcal{T}_{op}^{-T}\left(\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{p_j}^{\odot^T}\boldsymbol{p_j}^{\odot}\right)\mathcal{T}_{op}^{-1}$

证： 已知 $\boldsymbol{z_j}=\boldsymbol{T}_{op}\boldsymbol{p_j}$

等式左边 $=\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{T}_{op}\boldsymbol{p_j})^{\odot^T}(\boldsymbol{T}_{op}\boldsymbol{p_j})^{\odot}$ (2-1)

又 $\because (\boldsymbol{T}_{op}\boldsymbol{p_j})^{\odot^T}(\boldsymbol{T}_{op}\boldsymbol{p_j})^{\odot}\equiv\mathcal{T}_{op}^{-T}\boldsymbol{p_j}^{\odot^T}\boldsymbol{p_j}^{\odot}\mathcal{T}_{op}^{-1}$

式（2-1）$=$ 等式右边

问3：请证明 $\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z_j}^{\odot^T}(\boldsymbol{y_j}-\boldsymbol{z_j}) = \begin{bmatrix} \boldsymbol{y}-\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op}) \\ \boldsymbol{b}-\boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op}) \end{bmatrix}$

证：根据 $\boldsymbol{y}_j = \begin{bmatrix} \boldsymbol{y_j} \\ 1 \end{bmatrix}$，$\boldsymbol{p}_j = \begin{bmatrix} \boldsymbol{p}_j \\ 1 \end{bmatrix}$，$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{C}_{op} & -\boldsymbol{C}_{op}\boldsymbol{r}_{op} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$，$\boldsymbol{z_j} = \boldsymbol{T}_{op}\boldsymbol{p_j}$ 得：

$$\boldsymbol{z}_j = \begin{bmatrix} \boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op}) \\ 1 \end{bmatrix} \Longrightarrow \boldsymbol{z}_j^{\odot^T} = \begin{bmatrix} 1 & \boldsymbol{0}^T \\ (\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op}))^{\wedge} & \boldsymbol{0}^T \end{bmatrix}$$

$$\boldsymbol{z_j}^{\odot^T}(\boldsymbol{y_j}-\boldsymbol{z_j}) = \begin{bmatrix} \boldsymbol{y}_j-\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op}) \\ -\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op}) \end{bmatrix}$$

下面分两部分分布证明：

- $\frac{1}{w}\sum_{j=1}^{M} w_j\Big(\boldsymbol{y}_j-\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op})\Big) = \boldsymbol{y}-\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})$

  这一部分是显然成立的，只需要将 $\boldsymbol{y} = \frac{1}{w}\sum_{j=1}^{M} w_j\boldsymbol{y}_j$，$\quad \boldsymbol{p} = \frac{1}{w}\sum_{j=1}^{M} w_j\boldsymbol{p}_j$，$\quad w = \sum_{j=1}^{M} w_j$ 带入即可。

- $\frac{1}{w}\sum_{j=1}^{M} w_j\Big(-\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op})\Big) = \boldsymbol{b}-\boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})$

  已知 $\boldsymbol{b}_i = [\mathrm{tr}\,(\boldsymbol{1}_i^{\wedge}\boldsymbol{C}_{op}\boldsymbol{W}^T)]_i$

  $\boldsymbol{W} = \frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j-\boldsymbol{y})(\boldsymbol{p}_j-\boldsymbol{p})^T$

  $\boldsymbol{b}_i = \boldsymbol{1}_i^T\Big(-\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j-\boldsymbol{y})^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{p})\Big)$

  $\frac{1}{w}\sum_{j=1}^{M} w_j\Big(-\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op})\Big) = \frac{1}{w}\sum_{j=1}^{M} w_j\Big(-\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{r}_{op}+\boldsymbol{p}-\boldsymbol{p})\Big)$

  $= \frac{1}{w}\sum_{j=1}^{M} w_j\Big(-\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})-\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{p})\Big)$

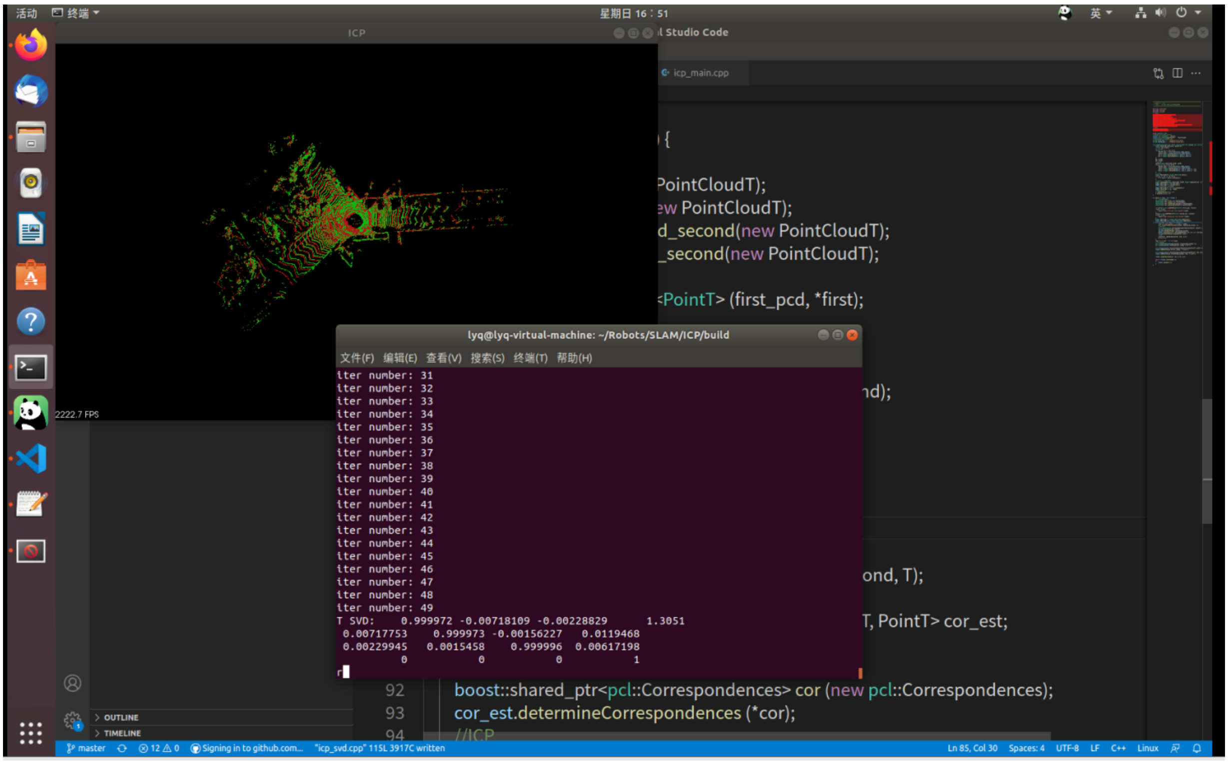
  $= \frac{1}{w}\sum_{j=1}^{M} w_j\Big(-\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})-(\boldsymbol{y}-\boldsymbol{y}+\boldsymbol{y}_j)^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{p})\Big)$

  $= \frac{1}{w}\sum_{j=1}^{M} w_j\Big(-\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})-(\boldsymbol{y}-\boldsymbol{y}_j)^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{p})-\boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{p})\Big)$

  $= \boldsymbol{b}-\boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})$

问4：写一段 ICP 程序，完成两个 ICP 文件的Pose 计算，不允许使用 PCL 或第三方点云库。

- 本人并不是搞感知方向的，所以程序参考了网上的代码；
- 并不是完全不使用 **PCL** 点云库，点云的读取和加载等部分都是使用点云库；
- 采用线性代数方法（SVD）求解
- https://zhuanlan.zhihu.com/p/104735380 三维点云配准 -- ICP 算法原理及推导



- 代码 `icp_svd.cpp` 如下：

```
/*********************************************************************
 * author: 李小山
 *  note    : 程序参考网上他人代码编写
 *******************************************************************/

#include <iostream>
#include <string>
#include <chrono>
// PCL库
#include <boost/make_shared.hpp>
#include <pcl/io/pcd_io.h>
#include <pcl/point_types.h>
#include <pcl/registration/icp.h>
#include <pcl/visualization/pcl_visualizer.h>
#include <pcl/features/fpfh_omp.h>
#include <pcl/features/fpfh.h>
#include <pcl/registration/correspondence_estimation.h>
#include <pcl/common/transforms.h>
//Eigen
#include <Eigen/Core>
#include <Eigen/Dense>

using namespace std;
//取别名，方便后面程序编写
typedef pcl::PointXYZ     PointT;
typedef pcl::PointCloud<PointT>     PointCloudT;
//PCD数据文件，注意文件路径
string first_pcd  = "../PCDdata/first.pcd";
string second_pcd = "../PCDdata/second.pcd";

void icp(PointCloudT::Ptr first, PointCloudT::Ptr second, pcl::Correspondences&
cor,Eigen::Matrix4d& T ){
    Eigen::Vector3d p1(0,0,0), p2(0,0,0);
    int N = cor.size();
    //计算质心
    for(int i = 0; i < N; i++){
        PointT pts1 = first->at(cor[i].index_query);
        PointT pts2 = second->at(cor[i].index_match);
        p1 += Eigen::Vector3d(pts1.x, pts1.y, pts1.z);
        p2 += Eigen::Vector3d(pts2.x, pts2.y, pts2.z);
```

```
    }
    p1 = p1/N;
    p2 = p2/N;
    //去质心
    vector<Eigen::Vector3d> q1(N), q2(N);
    for(int i = 0; i< N; i++){
        PointT pts1 = first->at(cor[i].index_query);
        PointT pts2 = second->at(cor[i].index_match);
        q1[i] = Eigen::Vector3d(pts1.x, pts1.y, pts1.z) - p1;
        q2[i] = Eigen::Vector3d(pts2.x, pts2.y, pts2.z) - p2;
    }
    //计算W
    Eigen::Matrix3d W = Eigen::Matrix3d::Zero();
    for(int i = 0; i < N; i++){
        W += q1[i] * q2[i].transpose();
    }
    // 对 W 进行 SVD 分解
    Eigen::JacobiSVD<Eigen::Matrix3d> svd(W, Eigen::ComputeFullU | Eigen::ComputeFullV);
    Eigen::Matrix3d U = svd.matrixU();
    Eigen::Matrix3d V = svd.matrixV();
    // 计算选择矩阵和平移向量
    Eigen::Matrix3d R = U * (V.transpose());
    Eigen::Vector3d t = p1 - R*p2;
    //得到变换矩阵
    T .topLeftCorner(3,3) = R;
    T.block(0,3,3,1) = t;
}

int main(int argc, char **argv) {
    //加载pcd数据文件
    PointCloudT::Ptr first (new PointCloudT);
    PointCloudT::Ptr second (new PointCloudT);
    PointCloudT::Ptr tranformed_second(new PointCloudT);
    PointCloudT::Ptr optimized_second(new PointCloudT);

    int err=pcl::io::loadPCDFile<PointT> (first_pcd, *first);
    if(err != 0){
        cout<<"load first_pcd file failed!"<<endl;
    }
    err=pcl::io::loadPCDFile<PointT> (second_pcd, *second);
    if(err != 0){
        cout<<"load second_pcd file failed!"<<endl;
    }
    Eigen::Matrix4d T = Eigen::Matrix4d::Identity();
    Eigen::Matrix4d d_T = Eigen::Matrix4d::Identity();
    for(int iter = 0; iter < 50; iter++){
        cout << "iter number: " << iter << endl;
        pcl::transformPointCloud(*second, *optimized_second, T);
        //点云匹配
        pcl::registration::CorrespondenceEstimation<PointT, PointT> cor_est;
        cor_est.setInputCloud (first);
        cor_est.setInputTarget (optimized_second);
        boost::shared_ptr<pcl::Correspondences> cor (new pcl::Correspondences);
        cor_est.determineCorrespondences (*cor);
        //ICP
        icp(first, optimized_second, *cor, d_T);
        T= d_T * T;
    }
    cout << "T SVD: " << T << endl;
    //结果可视化
    pcl::transformPointCloud(*second, *tranformed_second, T);
    pcl::visualization::PCLVisualizer viewer ("ICP");

    pcl::visualization::PointCloudColorHandlerCustom<PointT> green (first, 20, 180, 20);
    viewer.addPointCloud (first, green, "cloud1");
```

```
    pcl::visualization::PointCloudColorHandlerCustom<PointT> red (tranformed_second, 180, 20, 
20);
    viewer.addPointCloud (tranformed_second, red, "cloud2");

    viewer.setBackgroundColor (0.0, 0.0, 0.0);

    while (!viewer.wasStopped ())
    {
        viewer.spinOnce ();
    }
}
```