

cd 02_answer 运行如下命令；
source /home/rainlord/02_answer/devel/setup.bash

roslaunch lidar_localization front_end.launch //使用此命令运行 ros 程序

rosbag play bag.name //使用此命令播放 kitti bag 数据集

使用 icp 进行点云匹配，使用 rviz 进行显示；
关键帧点云数据已经转换到了第一帧点云对应的坐标系下，关键帧点云 pcd 文件存储在以下目录：
/home/rainlord/02_answer/src/lidar_localization/slam_data/key_frames

以下采用解析法求解 icp，对点云进行 icp 匹配；源码文件位于：
src/lidar_localization/src/models/registration/icp_svd_registration.cpp

//xs 是目标点云，ys 是源点云(当前帧点云)，求得的是 ys 到 xs 的位姿

```
void ICPSVDRegistration::GetTransform(  
const std::vector<Eigen::Vector3f> &xs,//pts1  
const std::vector<Eigen::Vector3f> &ys,//pts2  
Eigen::Matrix4f &transformation_  
) {  
const size_t N = xs.size();
```

// TODO -- find centroids of mu_x and mu_y:

```
Eigen::Vector3f xs_m={0.0f,0.0f,0.0f};//p1  
Eigen::Vector3f ys_m={0.0f,0.0f,0.0f};//p2  
for(int i=0;i<N;i++){  
xs_m += xs[i];  
ys_m += ys[i];  
}  
std::cout<<"xs_m:"<<xs_m<<std::endl;  
xs_m = xs_m/N;  
ys_m = ys_m/N;  
std::cout<<"xs_m after /N:"<<xs_m<<std::endl;  
std::vector<Eigen::Vector3f>xs_q(N),ys_q(N);//去除中心的点数据  
  
std::cout<<"xs[0]"<<xs[0]<<std::endl;  
for(int i=0;i<N;i++){
```

```

xs_q[i] = xs[i] - xs_m;
ys_q[i] = ys[i] - ys_m;
}
std::cout<<"xs_q[0] after -xs_m"<<xs_q[0]<<std::endl;
// TODO -- build H:
Eigen::Matrix3f H = Eigen::Matrix3f::Zero();
for(int i=0;i<N;i++){
H +=xs_q[i]*ys_q[i].transpose() ;
}
// TODO -- solve R:
Eigen::JacobiSVD<Eigen::Matrix3f> svd(H,Eigen::ComputeFullU |
Eigen::ComputeFullV);
Eigen::Matrix3f U =svd.matrixU();
Eigen::Matrix3f V =svd.matrixV();
Eigen::Matrix3f R_=U*(V.transpose());
if(R_.determinant()<0){
R_=-R_;
}
// TODO -- solve t:
Eigen::Vector3f t_ = xs_m-R_*ys_m;

//Eigen::Matrix3f R_V = R_.inverse();
//Eigen::Vector3f t_v = - R_V*t_;
// set output:
transformation_.setIdentity();
transformation_.block<3,3>(0,0)=R_;
transformation_.block<3,1>(0,3)=t_;
}

```

