DSLAM后端服务器使用说明

接收端程序在163服务器/home/caojf/DSLAM\_zu9\_proj/build/server.cpp

在/home/caojf/DSLAM\_zu9\_proj/build 路径下输入./server开启程序

在/home/caojf/DSLAM\_zu9\_proj输入vim examples/server.cpp 可以修改程序

发送端1程序在163服务器/home/caojf/DSLAM\_zu9\_proj/build/main.cpp

在/home/caojf/DSLAM\_zu9\_proj 路径下输入./build/main开启程序

在/home/caojf/DSLAM\_zu9\_proj输入vim examples/main.cpp 可以修改程序

修改源数据集 数据集路径在/home/caojf/DSLAM\_zu9\_proj/testdata/

发送端2程序在158服务器/home/linaro/DSLAM\_zu9\_proj/build/main.cpp

在/home/linaro/DSLAM\_zu9\_proj 路径下输入./build/main开启程序

在/home/linaro/DSLAM\_zu9\_proj输入vim examples/main.cpp 可以修改程序

修改源数据集 数据集路径在/home/linaro/DSLAM\_zu9\_proj/testdata/

工程文件在192.168.31.163服务器的/home/caojf/server 路径下；

头文件是 serverA.m文件;

ServerA文件是在netvlad匹配上之后未根据g2o优化的结果，直接根据真值轨迹进行了轨迹合并。（实际上进行了优化但结果不对）

测试时先运行头文件，然后再用163和158服务器里的发送端向168的接受端发送数据，再由后端读取，相当于是每200ms向后端发送一次一帧的pose和netvlad；

服务器输入变量a和a\_；a读取的是pose\_1当前帧中、的数据；a\_读取的是pose\_0中的数据；

输入变量b和d; b读取的是对应a读取的帧数的真值pose; d读取的是对应a\_读取的帧数的真值pose;

K1用来存储读取到的pose\_1中的帧数

K2用来存储读取到的pose\_0中的帧数

y1用来存储读取到的pose\_1中的位姿矩阵；（向量形式，sim\_o\_c存储矩阵形式）

y2用来存储读取到的pose\_0中的位姿矩阵；

z1用来存储读取到的netvlad\_1的向量；

z2用来存储读取到的netvlad\_0的向量；

画出的图像中蓝色的实线对应pose\_1的轨迹，蓝色的虚线对应其的真值轨迹；红色实线对应pose\_0的轨迹；

用接收的数据生成的.g2o文件在/home/caojf/server/dgs\_data/1 下 0.g2o对应pose\_1

1.g2o对应pose\_0；

优化后的g2o文件也在该文件夹下，读取优化后的g2o文件生成的新的优化后轨迹姿存储在decentr\_state和decentr\_state1中，decentr\_state是pose\_1优化后的结果decentr\_state1是pose\_0优化后的结果；