

源代码分析

源代码采用了 `vector<Sophus::SE3d, Eigen::aligned_allocator<Sophus::SE3d>>` 这一奇怪的数据形式,通过查阅eigen的文档,明白了这是为了保证存储eigen对象的vector的每一个元素都有固定的存储空间,从而避免了存储空间的错乱,更深层的原因是因为当用传值的方式把eigen对象以传值的方式传入vector时,其对齐信息会丢失.这算是eigen和stl容器交互时的一个bug.

The reason is that these **Eigen** objects have alignment modifiers that aren't respected when they are passed by value.

但是,在文档中提到了,在c++11 的版本之后,这个问题已经解决了,所以这种写法是没有必要的.因此在作业中,在vector中直接存储Sophus::SE3d的数据类型

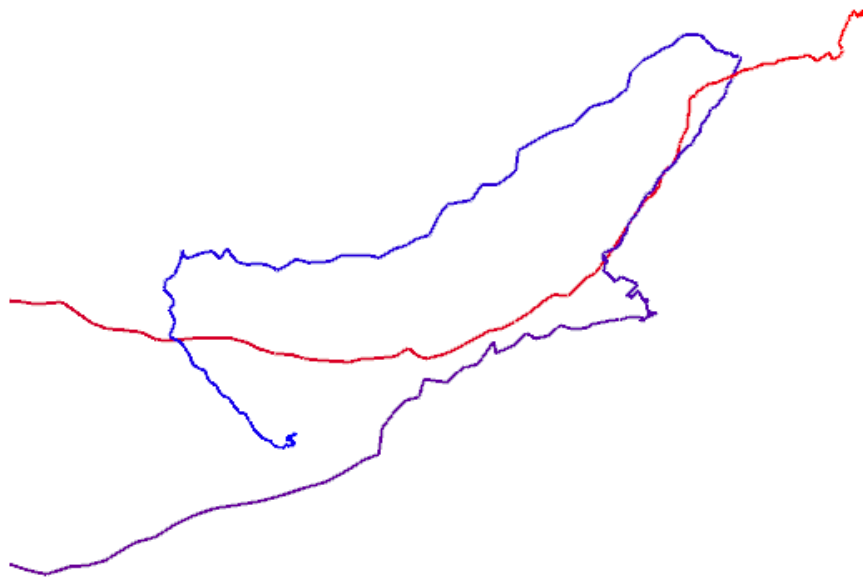
此外,源代码的构造临时对象.然后push back到vector的方法被修改为直接使用 `emplace_back`,提高效率.

参考资料:

[Using STL Containers with Eigen](#)

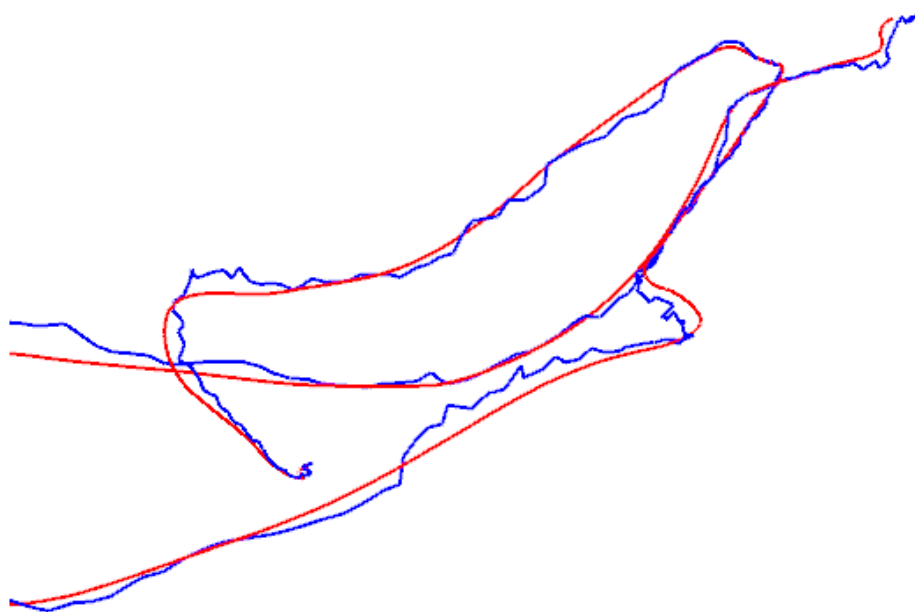
[Eigen::aligned_allocator< T > Class Template Reference](#)

轨迹效果



RMSE 误差

RMSE = 2.20728



真值和估计值对比效果