

## 一、代码调用分析：

调用 SetGNSSPose 函数设置 gnss pose 为初始位姿

```
bool MatchingFlow::UpdateMatching() {
    if (!matching_ptr_->HasInited()) {
        //
        // TODO: implement global initialization here
        //
        matching_ptr_->SetGNSSPose(current_gnss_data_.pose);
        // matching_ptr_->SetScanContextPose(current_cloud_data_);
        // Hints: You can use SetGNSSPose & SetScanContextPose from matching.hpp
        //

        // naive implementation:
        // Eigen::Matrix4f init_pose = Eigen::Matrix4f::Identity();

        // matching_ptr_->SetInitPose(init_pose);
        // matching_ptr_->SetInited();
    }

    return matching_ptr_->Update(current_cloud_data_, laser_odometry_);
}
```

这样设置定位效果很差，后来查阅资料得知，GNSS 是经纬度，而 current\_gnss\_data\_.pose 是第一个 GNSS 为原点转换过的 xyz 坐标。在任意点经过转换后的 xyz 坐标接近(0,0,0)，因此要把 0s 时刻的第一帧 GNSS 经纬度永远作为原点，而不是把(0,0,0)作为原点，代码中的三个数值是 0s 的 GNSS 经纬度数值。

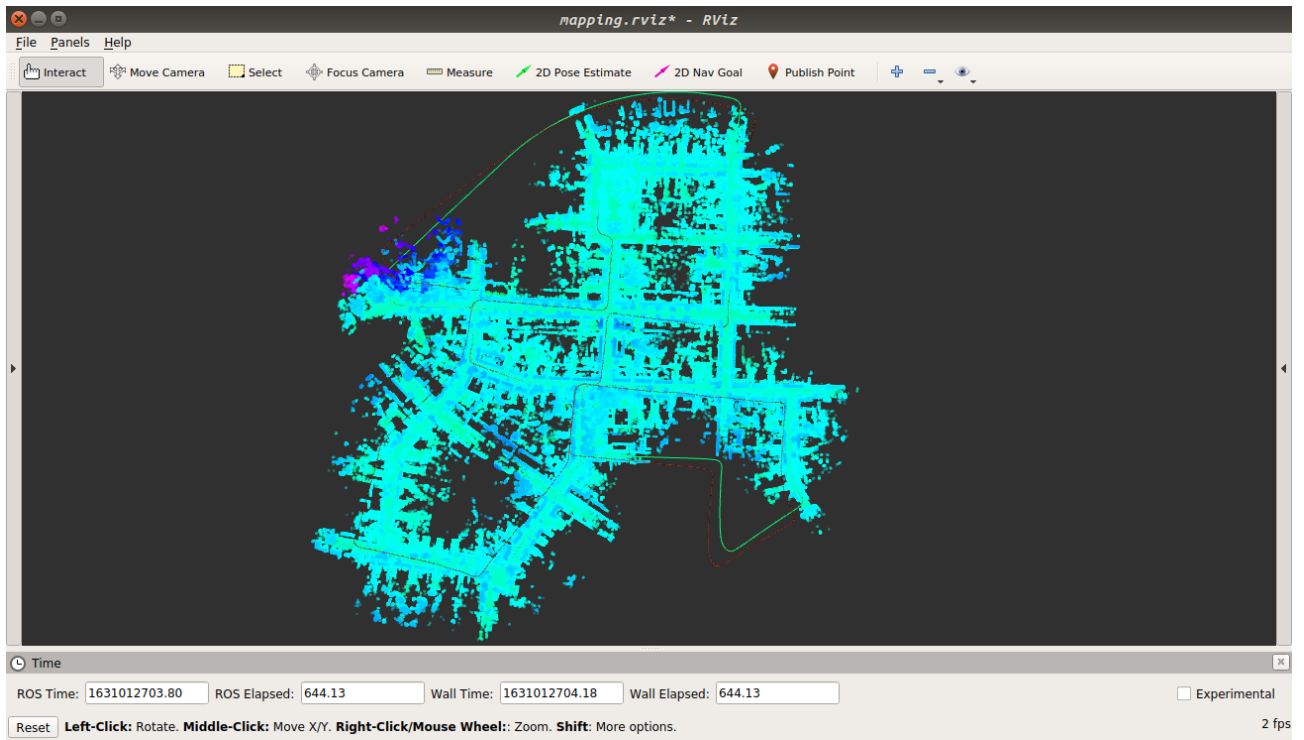
```
//静态成员变量必须在类外初始化
double lidar_localization::GNSSData::origin_longitude = 0.0;
double lidar_localization::GNSSData::origin_latitude = 0.0;
double lidar_localization::GNSSData::origin_altitude = 0.0;
bool lidar_localization::GNSSData::origin_position_inited = false;
GeographicLib::LocalCartesian lidar_localization::GNSSData::geo_converter;

namespace lidar_localization {
void GNSSData::InitOriginPosition() {
    // geo_converter.Reset(latitude, longitude, altitude);
    geo_converter.Reset(48.9827, 8.39046, 116.396);

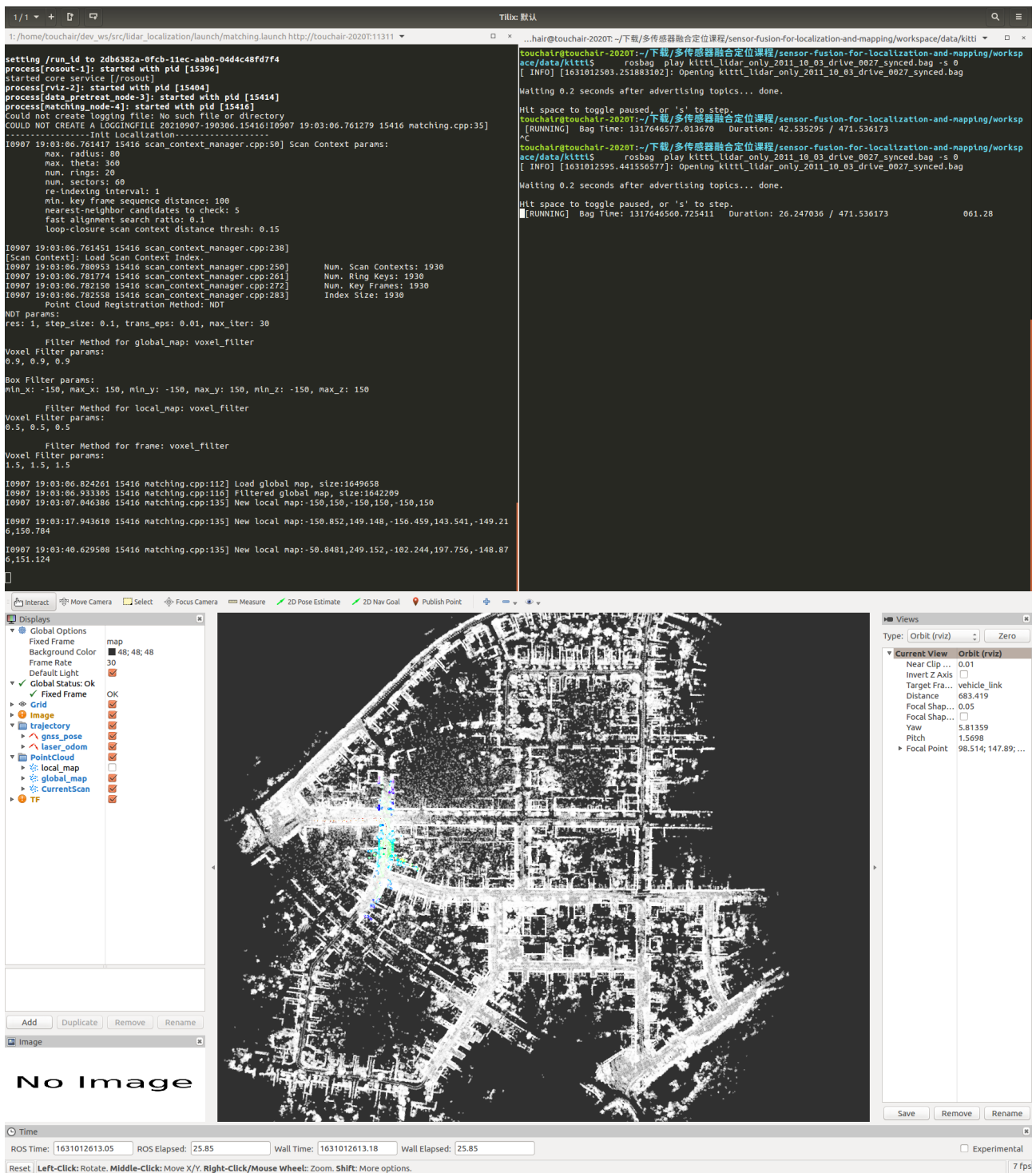
    origin_longitude = longitude;
    origin_latitude = latitude;
    origin_altitude = altitude;

    origin_position_inited = true;
}
```

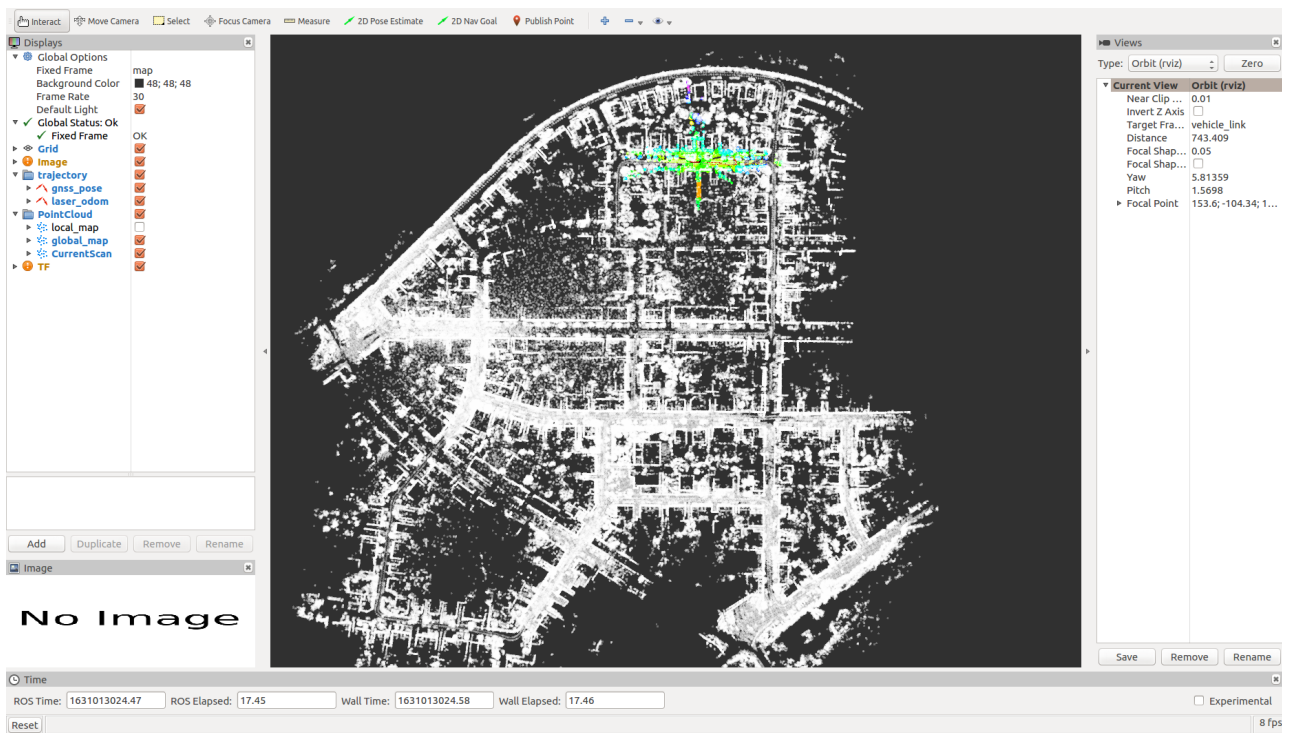
## 二、建图效果如下



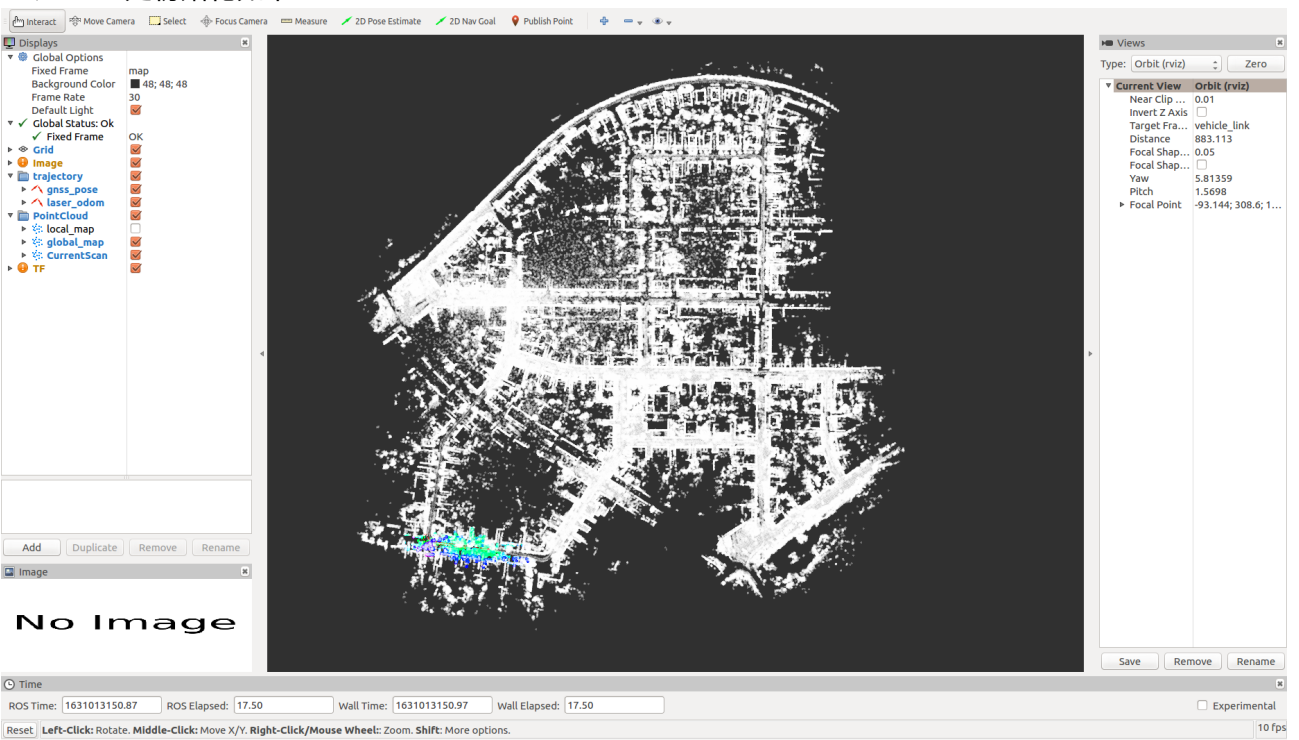
三、0s 处初始化如下



四、100s 处初始化如下

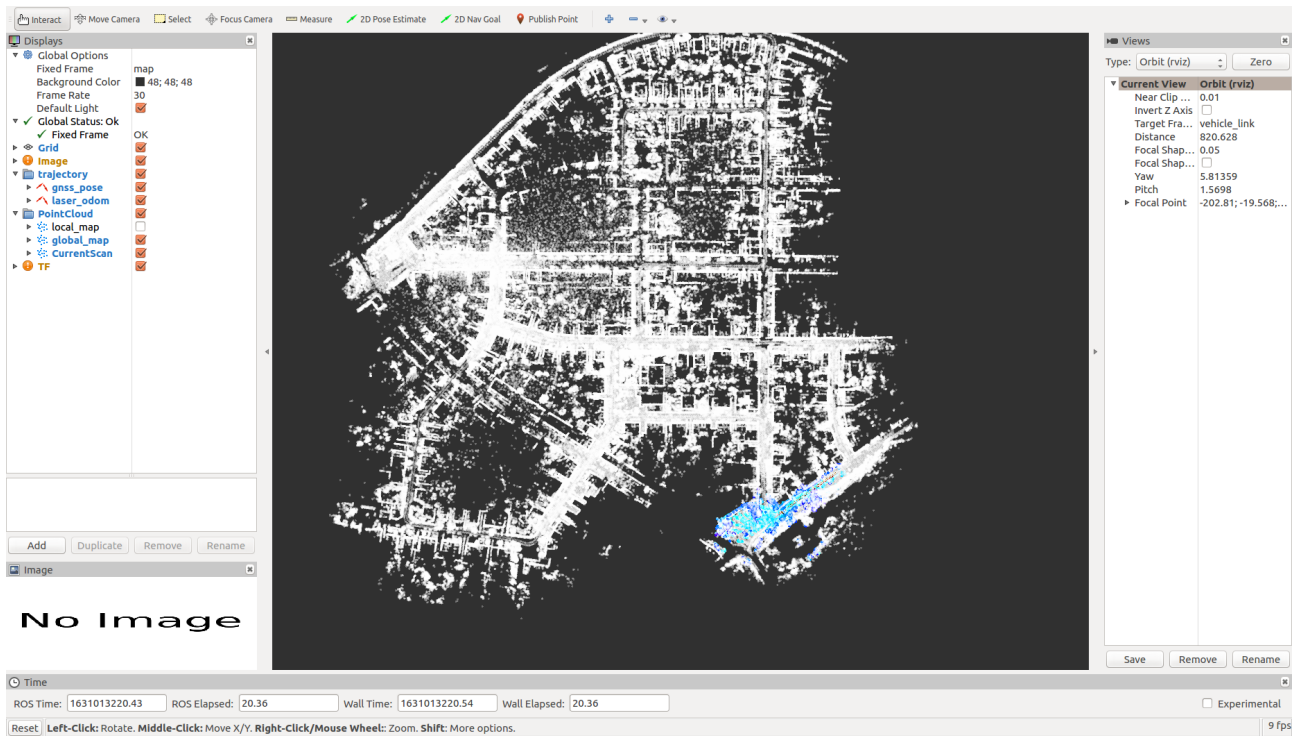


## 五、200s 处初始化如下



## 六、300s 处初始化如下





## 七、400s 处初始化如下

