

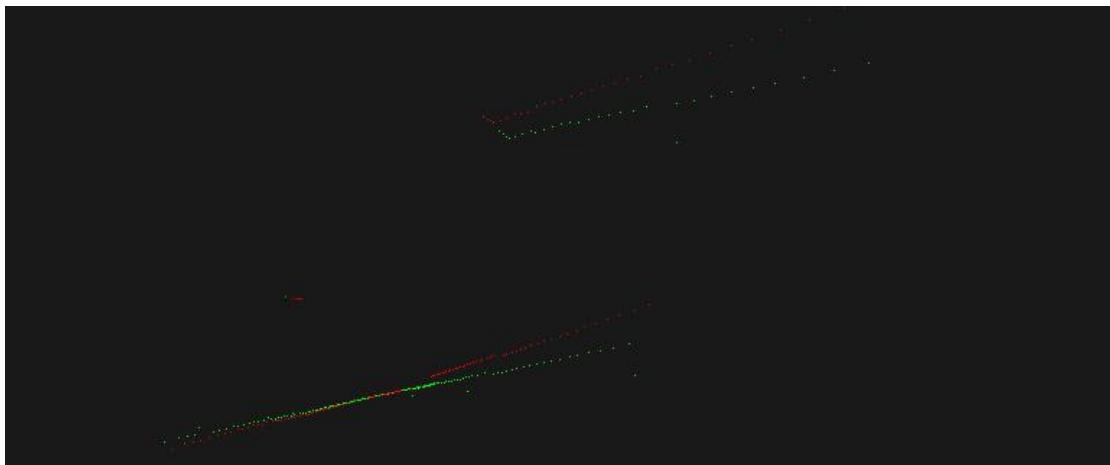
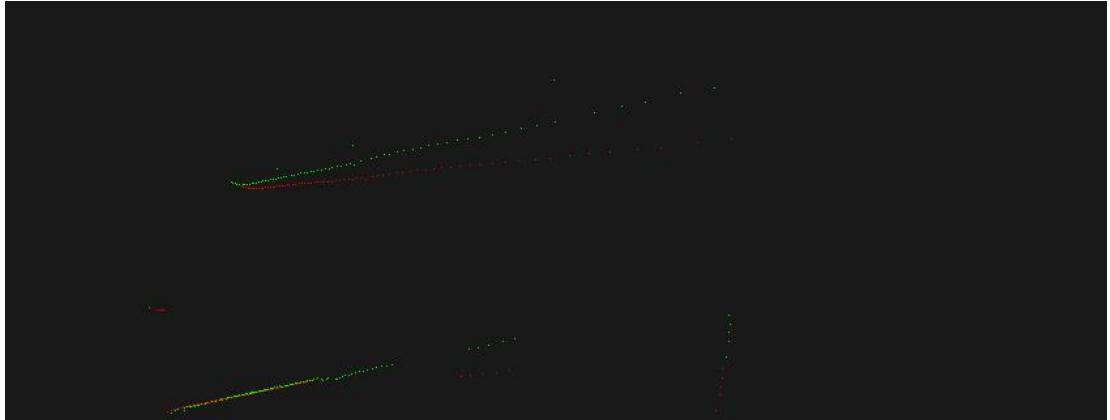
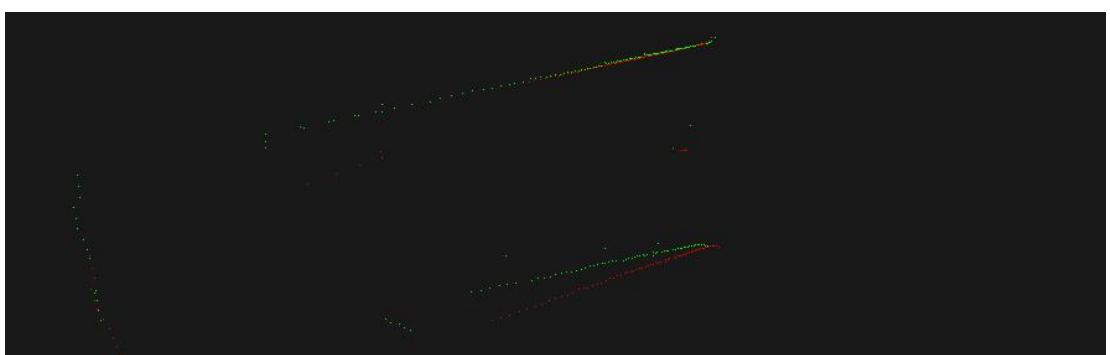
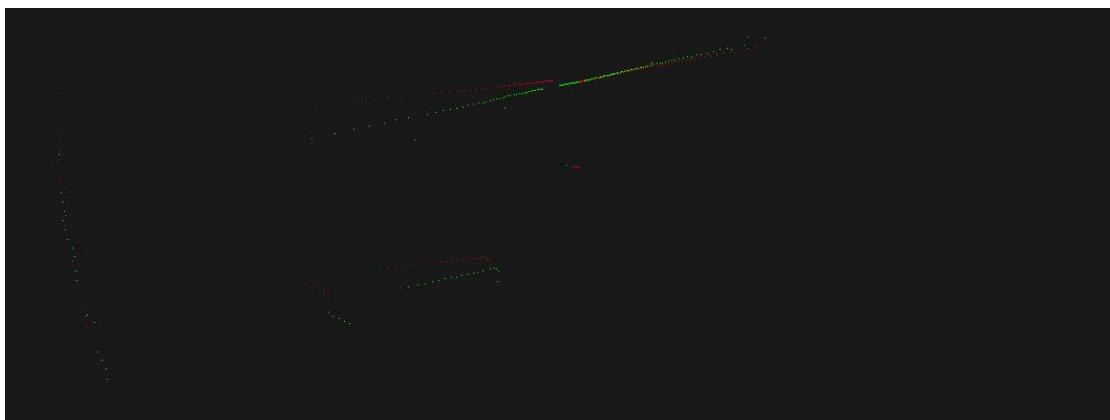
## 1. 补充去除激光雷达运动畸变模块的代码; (6 )

代码：

```
void Lidar_MotionCalibration(
    tf::Stamped<tf::Pose> frame_base_pose,
    tf::Stamped<tf::Pose> frame_start_pose,
    tf::Stamped<tf::Pose> frame_end_pose,
    std::vector<double>& ranges,
    std::vector<double>& angles,
    int startIndex,
    int& beam_number)
{
    //TODO
    //frame_base_pose.inverse;
    tf::Quaternion start_q = frame_start_pose.getRotation();
    tf::Quaternion end_q = frame_end_pose.getRotation();
    tf::Vector3 start_xy(frame_start_pose.getOrigin().getX(), frame_start_pose.getOrigin().getY(), 1);
    tf::Vector3 end_xy(frame_end_pose.getOrigin().getX(), frame_end_pose.getOrigin().getY(), 1);

    for (size_t i = startIndex; i < startIndex + beam_number; i++) {
        tf::Vector3 mid_xy = start_xy.lerp(end_xy, (i - startIndex) / (beam_number - 1));
        tf::Quaternion mid_q = start_q.slerp(end_q, (i - startIndex) / (beam_number - 1));
        tf::Transform mid_frame(mid_q, mid_xy);
        double x = ranges[i] * cos(angles[i]);
        double y = ranges[i] * sin(angles[i]);
        tf::Vector3 calib_point = frame_base_pose.inverse() * mid_frame * tf::Vector3(x, y, 1);
        ranges[i] = sqrt(calib_point[0] * calib_point[0] + calib_point[1] * calib_point[1]);
        angles[i] = atan2(calib_point[1], calib_point[0]);
    }
    //end of TODO
}
```

运行效果：



2. 阅读论文 Least-Squares Fitting of Two 3-D Points Sets, 推导并证明已知对应点的 ICP

求解方法；

推导：

对于两组点云：

$$X = x_1, x_2, \dots, x_{nx}$$

$$P = p_1, p_2, \dots, p_{np}$$

求解 R 和 t, 使得下式最小：

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - Rp_i - t\|^2$$

上式进行变换：

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - Rp_i - t - \mu_x + R\mu_p + \mu_x - R\mu_p\|^2$$

其中， $\mu_x$  和  $\mu_p$  分别为 X 点云和 P 点云的质心。

转换为去除质心的形式：

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - \mu_x - R(p_i - \mu_p) + (\mu_x - R\mu_p - t)\|^2$$

展开得到

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - \mu_x - R(p_i - \mu_p)\|^2 + \|(\mu_x - R\mu_p - t)\|^2 + 2(x_i - \mu_x - R(p_i - \mu_p))^T(\mu_x - R\mu_p - t)$$

由于后面两项都有

$$\mu_x - R\mu_p - t$$

我们可以令该项等于 0, 然后式子转化为

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - \mu_x - R(p_i - \mu_p)\|^2$$

该式子仅与  $R$  有关，可以寻找可以最小化上式的  $R$ ，然后根据  $\mu_x - R\mu_p - t = 0$  求解  $t$ 。

上式进一步转化，得到：

$$\frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i^* - Rp_i^*\|^2$$

其中， $X^*$  和  $P^*$  是原始点云减去对应质心的点点云， $x^*$  和  $p^*$  是其中的点。

$$X^* = \{x_i - \mu_x\} = x_i^*$$

$$P^* = \{p_i - \mu_p\} = p_i^*$$

展开得到：

$$\frac{1}{N_p} \sum_{i=1}^{N_p} x_i^{*T} x_i^1 + p_i^{*T} R^T R p_i^* - 2x_i^* R p_i^*$$

其中旋转矩阵  $R$  因为是正交矩阵，所以  $R^T R = 1$ ，因此求上式的最小值等价于求最后一项的最小值：

$$\sum_{i=1}^{N_p} -2x_i^{*T} R p_i^*$$

即该式的最大值：

$$\sum_{i=1}^{N_p} x_i^{*T} R p_i^*$$

进行转化，用矩阵的迹进行表示：

$$\sum_{i=1}^{N_p} \text{Trace}(R x_i^* p_i^{*T}) = \text{Trace}(RH)$$

其中，

$$H = \sum_{i=1}^{N_p} x_i^* p_i^{*T}$$

对 H 进行 SVD 分解得到：

$$H = U \Lambda V^T$$

设定一个变量 X 为

$$X = VU^T$$

得到：

$$XH = VU^T U \Lambda V^T = V \Lambda V^T$$

此时得到的 XH 是一个正定对称矩阵。

根据定理：

假设矩阵 A 为正定对称矩阵，则对于任意的正交矩阵 B，都有

$$\text{Trace}(A) \geq \text{Trace}(BA)$$

类比到我们的推导这里，我们准备求解可以让  $\text{Trace}(RH)$  最大的 R 值，我们发现，当  $R=VU^T$  的时候，RH 的结果是一个正定对称矩阵，而对于任意一个正定对称矩阵来说，根据定理可以知道，我们再乘以任何一个正交矩阵（旋转矩阵是正交矩阵），都会导致 Trace 值下降，所以说明，此时的  $R=VU^T$  取值可以让  $\text{Trace}(RH)$  取得最大值。

因此：

$$R = X = VU^T$$

$$t = \mu_x - R\mu_p$$

3. 阅读论文 Precise indoor localization for mobile laser scanner 前两章，回答问题。 (2 分)

(1) 根据第二章内容，简述激光雷达测距原理；

(2) 简要介绍一下右图的含义。

(1)

激光雷达测距原理分为两种：一种是根据三角法测量距离，另一种是测量飞行时间，即 TOF。

后者又分为两种测量方式，分别是直接测量飞行时间和测量相位差。

公式如下所示：

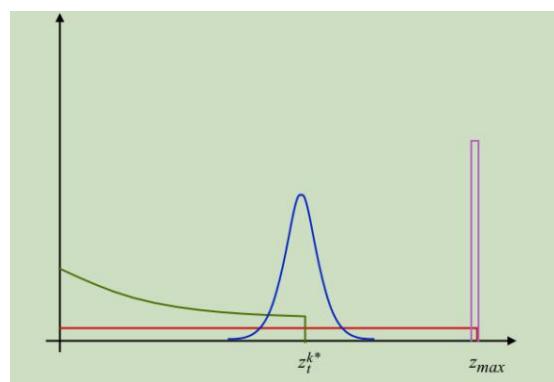
直接测量飞行时间：

$$r = \frac{\Delta t \cdot c}{n \cdot 2}.$$

测量相位差：

$$D = \frac{c \Delta \phi}{4\pi f_m}$$

(2)



右图是激光雷达光束模型的联合概率分布图。

其中  $Z_t^k$  表示测量返回的距离值， $Z_{max}$  表示激光雷达可能测量得到的最大数值。

由四种概率分布组成：

蓝色是高斯概率分布，代表实际测量距离及其不确定性。

绿色是指数分布，代表检测到动态障碍物的可行性。

红色代表随机噪声。

粉色代表错误测量的可能性。

4. 简答题，开放性答案：设计使用 IMU 去除激光雷达运动畸变的方法并回答问题。

(1) 仅用 IMU 去除运动畸变可能会有哪些不足之处？

(2) 在仅有 IMU 和激光雷达传感器的情况下，你会如何设计运动畸变去除方案(平移+旋转)，达到较好的畸变去除效果？

(1) 不足之处：

IMU 对于线加速度进行二次积分可能误差较大

(2) 设计：

在每一个激光雷达数据帧内部，使用 imu 进行积分，得到 IMU 里程计。

通过时间同步，然后确定每一个激光雷达点束的在 IMU 里程计坐标系下面的位置

将每一个激光雷达点束都矫正到该帧起始位姿下面。

对于 n 时刻矫正好的激光雷达数据，使用 icp 算法，将 n 时刻与 n-1 时刻的矫正好的激光雷达帧进行匹配，得到激光雷达匹配的位姿。

使用这个位姿纠正 imu 的累计误差，然后继续矫正下一帧数据。