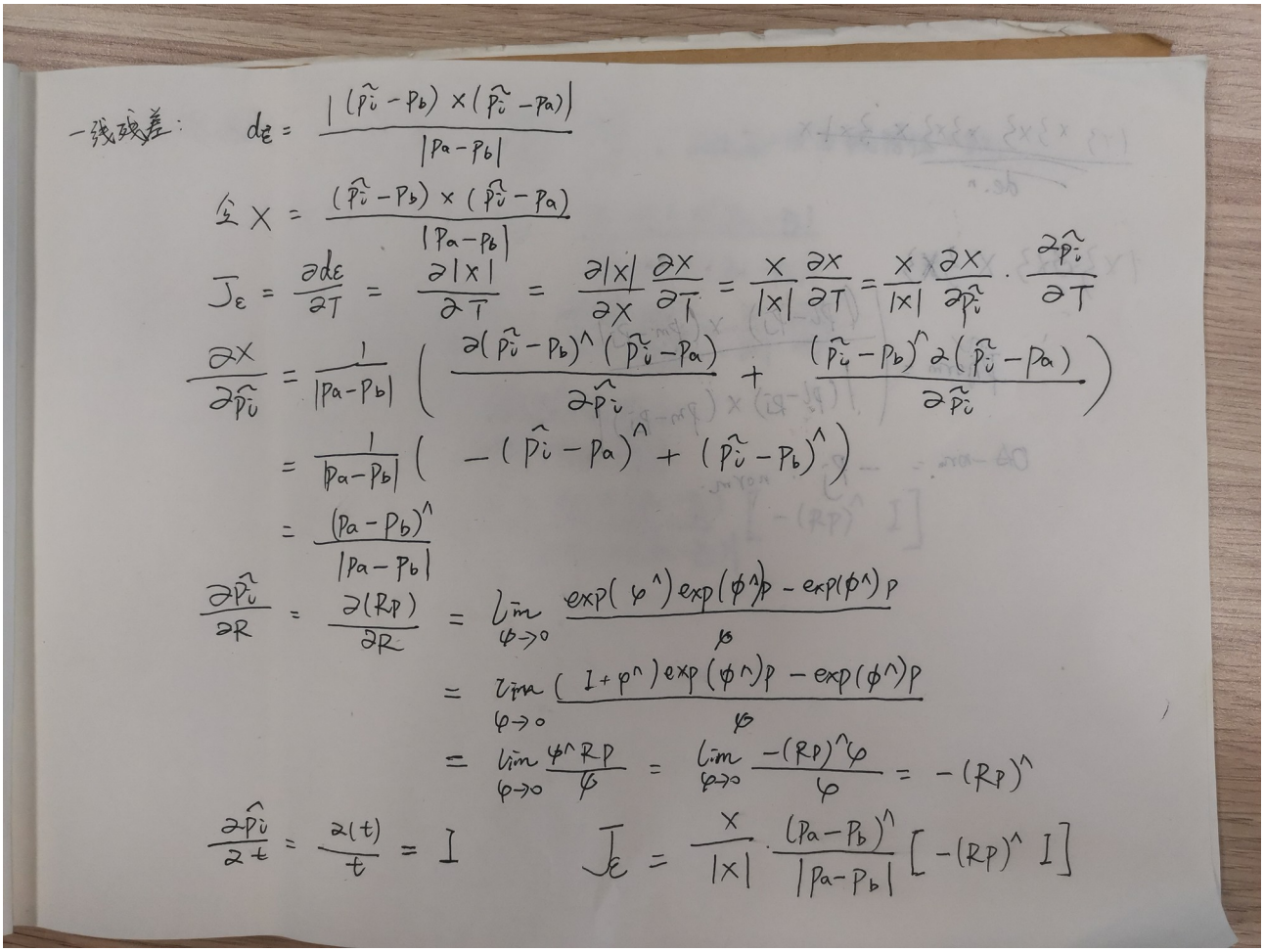
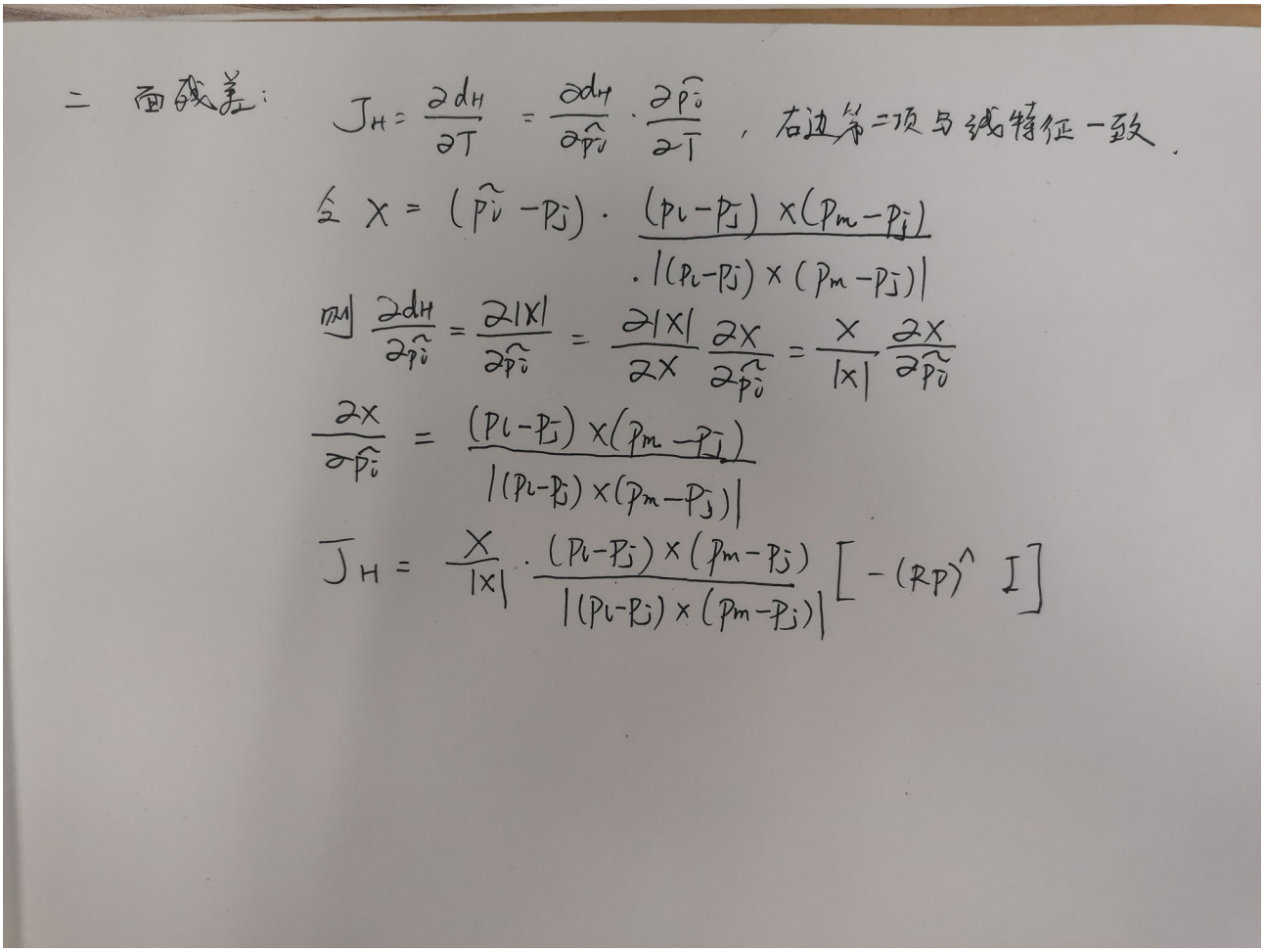
一、公式推导

线特征残差的雅可比



面特征残差的雅可比


二 面残差：
$J_H = \frac{\partial d_H}{\partial T} = \frac{\partial d_H}{\partial \hat{p}_i} \cdot \frac{\partial \hat{p}_i}{\partial T}$，右边第二项与线特征一致。
令 $X = (\hat{p}_i - p_j) \cdot \frac{(p_l - p_j) \times (p_m - p_j)}{|(p_l - p_j) \times (p_m - p_j)|}$
则 $\frac{\partial d_H}{\partial \hat{p}_i} = \frac{\partial |X|}{\partial \hat{p}_i} = \frac{\partial |X|}{\partial X} \frac{\partial X}{\partial \hat{p}_i} = \frac{X}{|X|} \frac{\partial X}{\partial \hat{p}_i}$
$\frac{\partial X}{\partial \hat{p}_i} = \frac{(p_l - p_j) \times (p_m - p_j)}{|(p_l - p_j) \times (p_m - p_j)|}$
$J_H = \frac{X}{|X|} \cdot \frac{(p_l - p_j) \times (p_m - p_j)}{|(p_l - p_j) \times (p_m - p_j)|} \left[-(Rp)^\wedge \; I \right]$

二、公式与代码的对应关系

线特征的残差和雅可比

```
virtual bool Evaluate(double const *const *parameters, double *residuals, double **jacobians) const
{
    Eigen::Map<const Eigen::Quaterniond> q_last_curr(parameters[0]);
    Eigen::Map<const Eigen::Vector3d> t_last_curr(parameters[0] + 4);

    Eigen::Vector3d lp;
    Eigen::Vector3d lp_r;
    lp = q_last_curr * curr_point + t_last_curr;
    lp_r = q_last_curr * curr_point;
    Eigen::Vector3d nu = (lp - last_point_b).cross(lp - last_point_a);
    Eigen::Vector3d de = last_point_a - last_point_b;
    Eigen::Vector3d re = last_point_b - last_point_a;

    // 对应线特征残差
    Eigen::Vector3d residualsVector;
    residualsVector << nu.x() / de.norm(), nu.y() / de.norm(), nu.z() / de.norm();
    residuals[0] = residualsVector.norm();

    if(jacobians != NULL)
    {
        if(jacobians[0] != NULL)
        {
            if(residuals[0] != 0)
            {
                nu = residualsVector / residuals[0];
            }

            Eigen::Matrix3d skew_re = skew(re);
            Eigen::Matrix3d skew_de = skew(de);
            Eigen::Matrix3d skew_lp_r = skew(lp_r);

            Eigen::Matrix<double, 3, 6> dp_by_so3;
            dp_by_so3.block<3,3>(0,0) = -skew_lp_r;
            (dp_by_so3.block<3,3>(0,3)).setIdentity();
            Eigen::Map<Eigen::Matrix<double, 1, 7, Eigen::RowMajor> > J_se3(jacobians[0]);

            J_se3.setZero();
            J_se3.block<1,6>(0,0) = nu.transpose() * skew_de * dp_by_so3 / de.norm();
        }
    }
```

$$d_{\varepsilon} = \frac{|(\tilde{P_i} - P_b) \times (\tilde{P_i} - P_a)|}{|P_a - P_b|}$$

$$X = \frac{(\tilde{P_i} - P_b) \times (\tilde{P_i} - P_a)}{|P_a - P_b|}$$

$$J_{\varepsilon} = \frac{X}{|X|} \cdot \frac{(P_a - P_b)^{\wedge}}{|P_a - P_b|} \left[-(RP)^{\wedge} \; I \right]$$

面特征的残差和雅可比

```
virtual bool Evaluate(double const *const *parameters, double *residuals, double **jacobians) const
{
    Eigen::Map<const Eigen::Quaterniond> q_last_curr(parameters[0]);
    Eigen::Map<const Eigen::Vector3d> t_last_curr(parameters[0] + 4);
    Eigen::Vector3d lp = q_last_curr * curr_point + t_last_curr;
    Eigen::Vector3d lp_r = q_last_curr * curr_point;

    Eigen::Vector3d de = (last_point_j - last_point_l).cross(last_point_j - last_point_m);
    de.normalize();

    double nu = (lp - last_point_j).dot(de);

    residuals[0] = nu;


    if(jacobians != NULL)
    {
        if(jacobians[0] != NULL)
        {

            if(residuals[0] != 0)
            {
                nu = nu / residuals[0];
            }

            Eigen::Matrix3d skew_lp_r = skew(lp_r);

            Eigen::Matrix<double, 3, 6> dp_by_so3;
            dp_by_so3.block<3,3>(0,0) = -skew_lp_r;
            (dp_by_so3.block<3,3>(0,3)).setIdentity();

            Eigen::Map<Eigen::Matrix<double, 1, 7, Eigen::RowMajor> > J_se3(jacobians[0]);

            J_se3.setZero();
            J_se3.block<1,6>(0,0) = de.transpose() * dp_by_so3;


        }
    }
}
```

$$X = (\tilde{P_i} - P_j) \cdot \frac{(P_l - P_j) \times (P_m - P_j)}{|(P_l - P_j) \times (P_m - P_j)|}$$

$$J_H = \frac{X}{|X|} \cdot \frac{(P_l - P_j) \times (P_m - P_j)}{|(P_l - P_j) \times (P_m - P_j)|} \left[-(RP)^{\wedge} \; I \right]$$

三、参数块和残差块的调用

参数块的调用

```
ceres::Problem::Options problem_options;

ceres::Problem problem(problem_options);
// problem.AddParameterBlock(para_q, 4, new PoseSE3Parameterization());
// problem.AddParameterBlock(para_t, 3);
problem.AddParameterBlock(parameters, 7, new PoseSE3Parameterization());
```

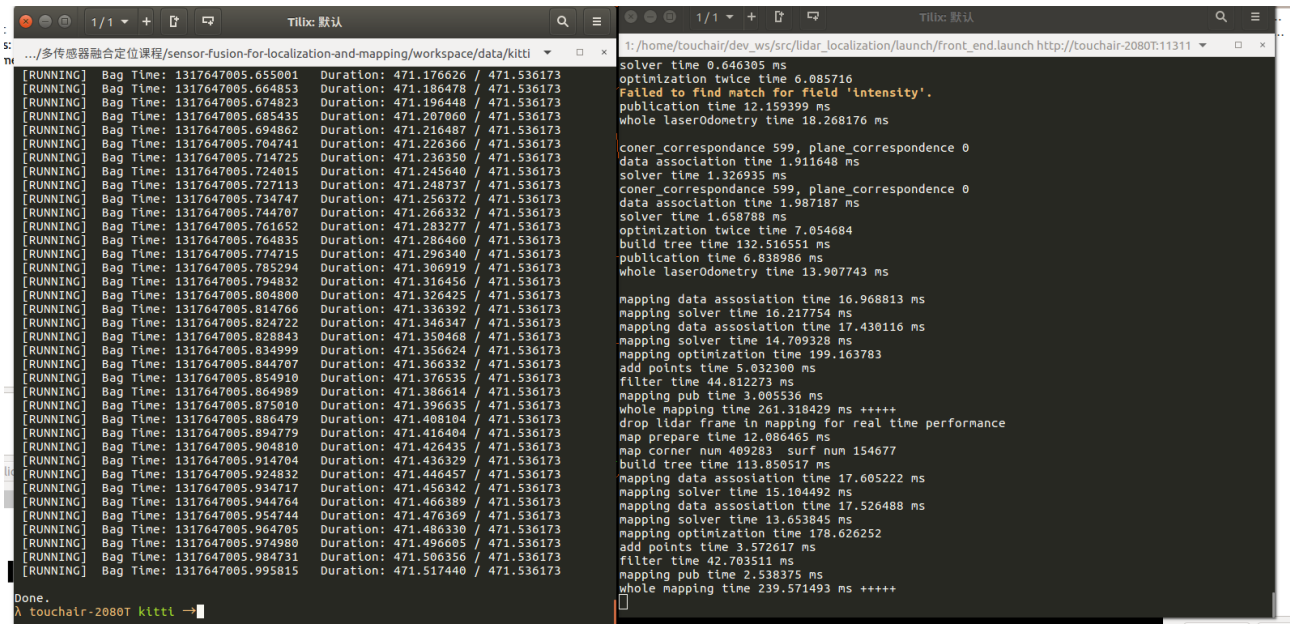
线残差块的调用

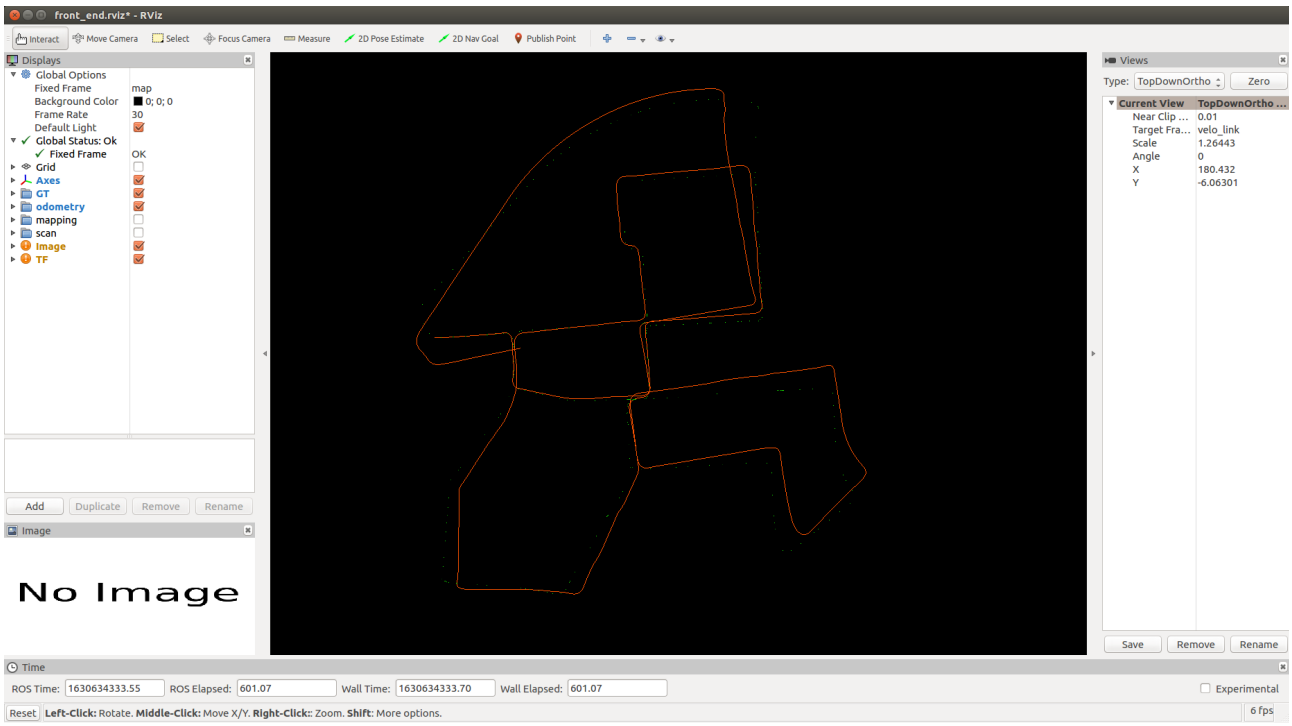
```
ceres::CostFunction *cost_function = new EdgeCostFunction(curr_point, last_point_a,
last_point_b, s);


// problem.AddResidualBlock(cost_function, loss_function, para_q, para_t);
problem.AddResidualBlock(cost_function, loss_function, parameters);
corner_correspondence++;
```

面残差块的调用

```
ceres::CostFunction *cost_function = new PlaneCostFunction(curr_point, last_point_a,
last_point_b, last_point_c, s);

// problem.AddResidualBlock(cost_function, loss_function, para_q, para_t);
problem.AddResidualBlock(cost_function, loss_function, parameters);
plane_correspondence++;
```

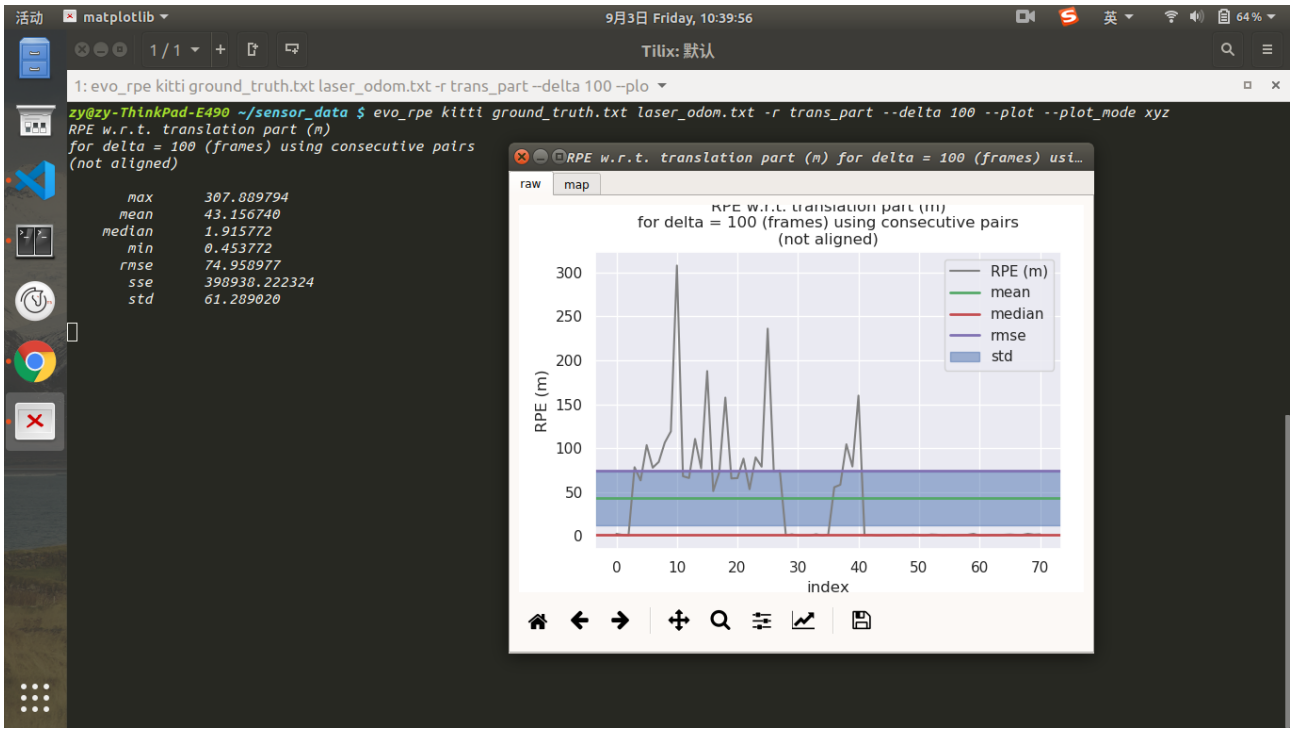
四、运行结果



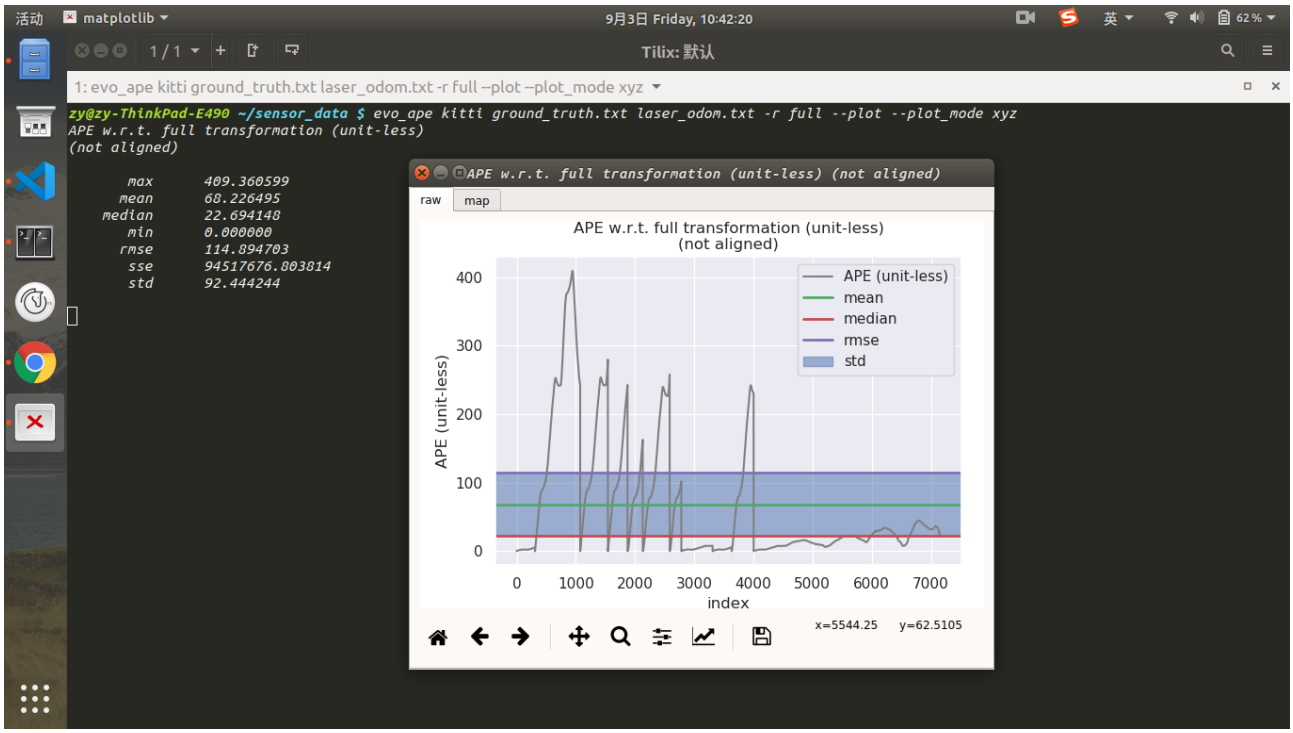

front_end.rviz* - RViz
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
0; 0; 0
Frame Rate
30
Default Light
Global Status: Ok
Fixed Frame
OK
Grid
Axes
GT
odometry
mapping
scan
Image
TF
Add
Duplicate
Remove
Rename
Image
No Image
Views
Type:
TopDownOrtho
Zero
Current View
TopDownOrtho ...
Near Clip ...
0.01
Target Fra...
velo_link
Scale
1.26443
Angle
0
X
180.432
Y
-6.06301
Save
Remove
Rename
Time
ROS Time:
1630634333.55
ROS Elapsed:
601.07
Wall Time:
1630634333.70
Wall Elapsed:
601.07
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
6 fps

五、精度评价

分段误差


活动
matplotlib
9月3日 Friday, 10:39:56
英
64 %
1 / 1
Tilix: 默认
1: evo_rpe kitti ground_truth.txt laser_odom.txt -r trans_part --delta 100 --plo
zy@zy-ThinkPad-E490 ~/sensor_data $ evo_rpe kitti ground_truth.txt laser_odom.txt -r trans_part --delta 100 --plot --plot_mode xyz
RPE w.r.t. translation part (m)
for delta = 100 (frames) using consecutive pairs
(not aligned)
max 307.889794
mean 43.156740
median 1.915772
min 0.453772
rmse 74.958977
sse 398938.222324
std 61.289020
RPE w.r.t. translation part (m) for delta = 100 (frames) usi...
raw
map
for delta = 100 (frames) using consecutive pairs
(not aligned)
RPE (m)
mean
median
rmse
std
RPE (m)
index
0
50
100
150
200
250
300
0
10
20
30
40
50
60
70

总体误差


活动
matplotlib
9月3日 Friday, 10:42:20
英
62 %
1 / 1
Tilix: 默认
1: evo_ape kitti ground_truth.txt laser_odom.txt -r full --plot --plot_mode xyz
zy@zy-ThinkPad-E490 ~/sensor_data $ evo_ape kitti ground_truth.txt laser_odom.txt -r full --plot --plot_mode xyz
APE w.r.t. full transformation (unit-less)
(not aligned)
max 409.360599
mean 68.226495
median 22.694148
min 0.000000
rmse 114.894703
sse 94517676.803814
std 92.444244
APE w.r.t. full transformation (unit-less) (not aligned)
raw
map
APE w.r.t. full transformation (unit-less)
(not aligned)
APE (unit-less)
mean
median
rmse
std
400
300
200
100
0
0
1000
2000
3000
4000
5000
6000
7000
index
x=5544.25
y=62.5105