Lab 5: Scan Matching

Yuwei Wu Christopher Hsu Peyman Norouzi

1. Approach Description

The code implements the PL-ICP method for the matching and use two strategies to improve the performance of matching.

Rather than match each point, which is quite slow and inefficient, the PL-ICP applies the distance of the line between two adjacent points rather than one point for matching.

For fast correspondence search, there are two improvements that can decrease the range of points to be checked each time. First is the local search with early termination. We can calculate the min_best_distance and terminate search when min_best_distance exceeds the current best distance, rather than search all points in lidar scan. The second strategy is to use the jump table for each scan point so that there is no need to traverse all the points in the local search but skip to the max/min point.

2. Roadblocks

2.1 The range of transform

When we do the transform, we need to range the theta in [-pi, pi), so we need to add: p_i_w.wrapTheta(); in our code.

2.2 The update in the transform.cpp

At first, the code in transform for curr_trans = Transform(x(0), x(1), theta); but it will make the iteration in the optimization part useless because the computation does not accumulate over time. Thus, we can accumulate the computed transform each time so it makes sense for the update step.

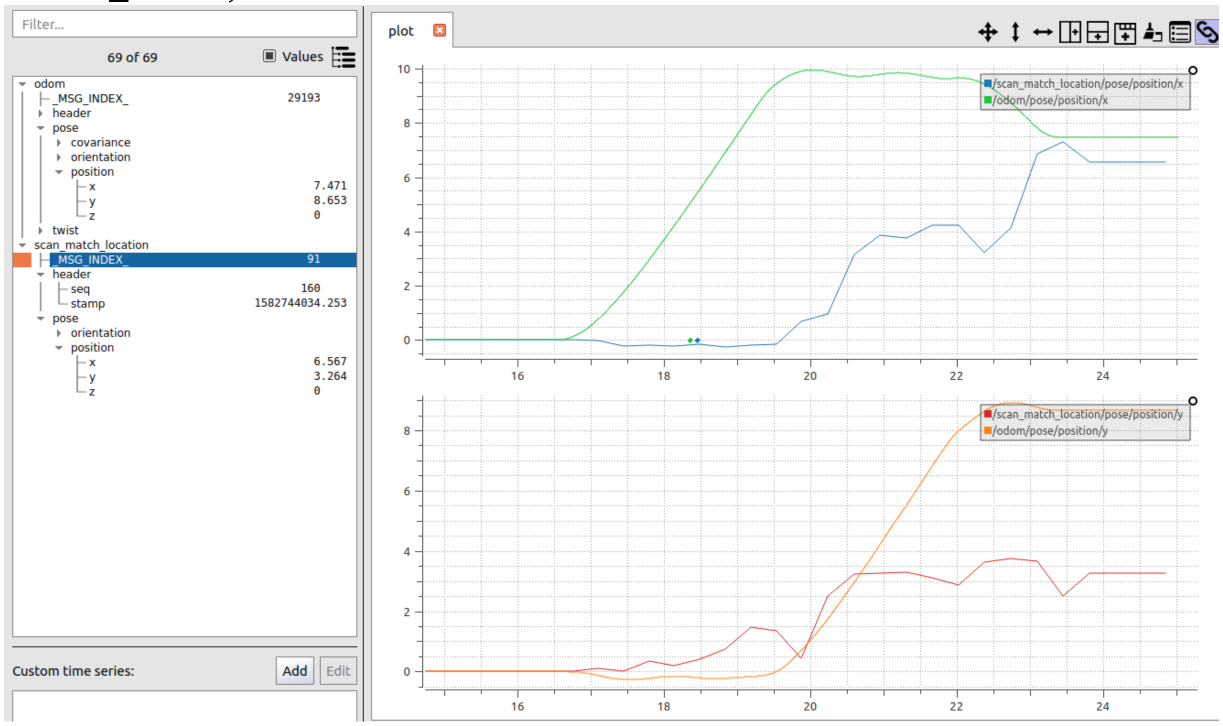
3.3 weights in Ci

Actually, in the paper, Ci = wi*ni*ni^T, so we consider whether we can change the weight to improve performance. After our tuning process, we decide to keep it as wi = 1

3. **Performance and Analysis**

We would like to tune the number of iterations because there is a trade-off between the number of iterations and the computation speed of it. The trade-off is due to the fact that each time the trajectory is not the same, we plot the relations of x and y compared to the real odometry.

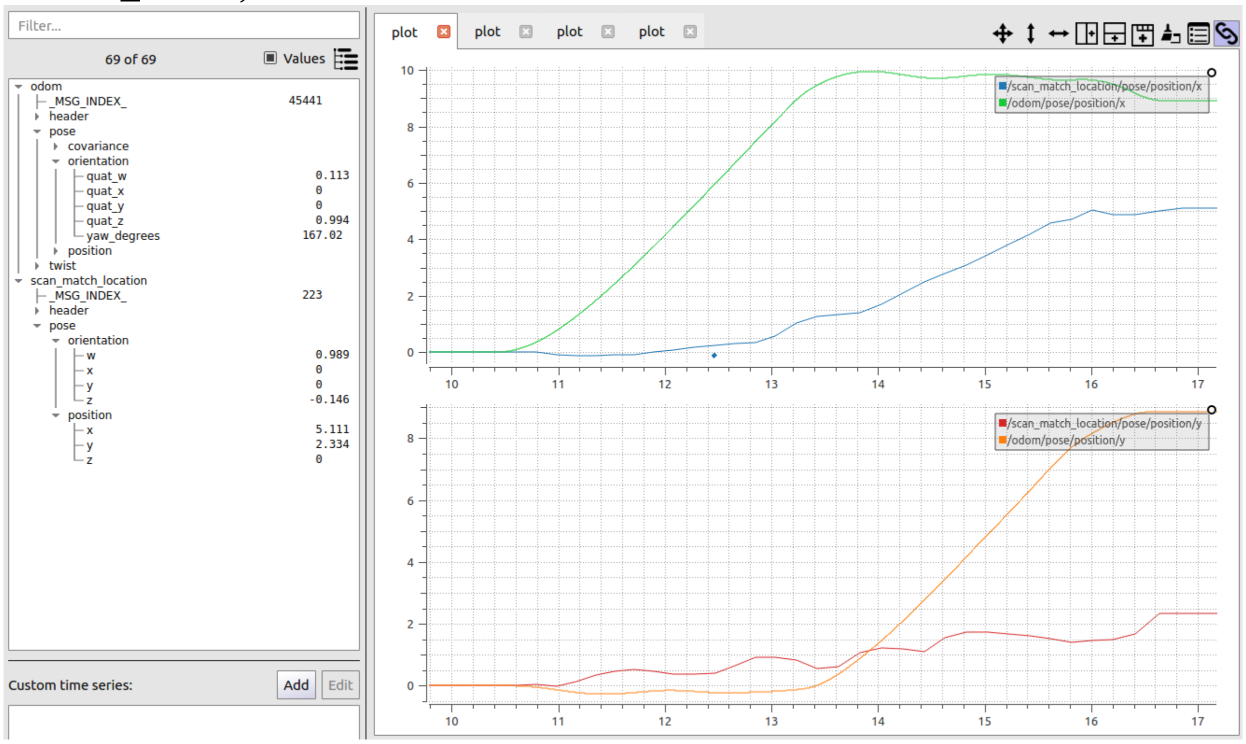
int number_iter =1;



```
Filter...
69 of 69                         Values
odom
 ├ _MSG_INDEX_                 29193
 ▸ header
 ▾ pose
   ▸ covariance
   ▸ orientation
   ▾ position
      ├ x                       7.471
      ├ y                       8.653
      └ z                       0
 ▸ twist
scan_match_location
 ├ _MSG_INDEX_                 91
 ▾ header
   ├ seq                       160
   └ stamp          1582744034.253
 ▾ pose
   ▸ orientation
   ▾ position
      ├ x                       6.567
      ├ y                       3.264
      └ z                       0
Custom time series:        Add   Edit
```

plot
/scan_match_location/pose/position/x
/odom/pose/position/x
/scan_match_location/pose/position/y
/odom/pose/position/y

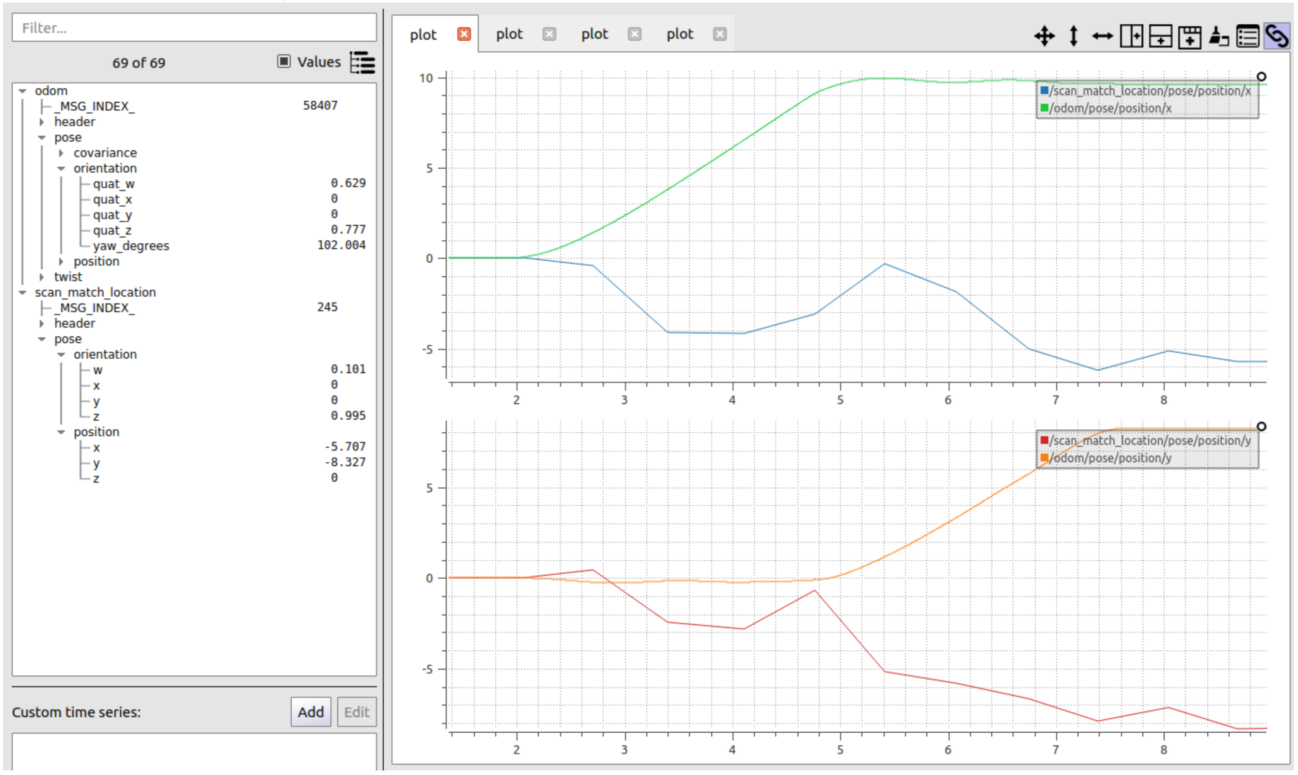
int number_iter =3;



```
Filter...
69 of 69                         Values
odom
 ├ _MSG_INDEX_                 45441
 ▸ header
 ▾ pose
   ▸ covariance
   ▾ orientation
      ├ quat_w                  0.113
      ├ quat_x                  0
      ├ quat_y                  0
      ├ quat_z                  0.994
      └ yaw_degrees             167.02
   ▸ position
 ▸ twist
scan_match_location
 ├ _MSG_INDEX_                 223
 ▸ header
 ▾ pose
   ▾ orientation
      ├ w                       0.989
      ├ x                       0
      ├ y                       0
      └ z                      -0.146
   ▾ position
      ├ x                       5.111
      ├ y                       2.334
      └ z                       0
Custom time series:        Add   Edit
```

plot
plot
plot
plot
/scan_match_location/pose/position/x
/odom/pose/position/x
/scan_match_location/pose/position/y
/odom/pose/position/y

int number_iter =10;

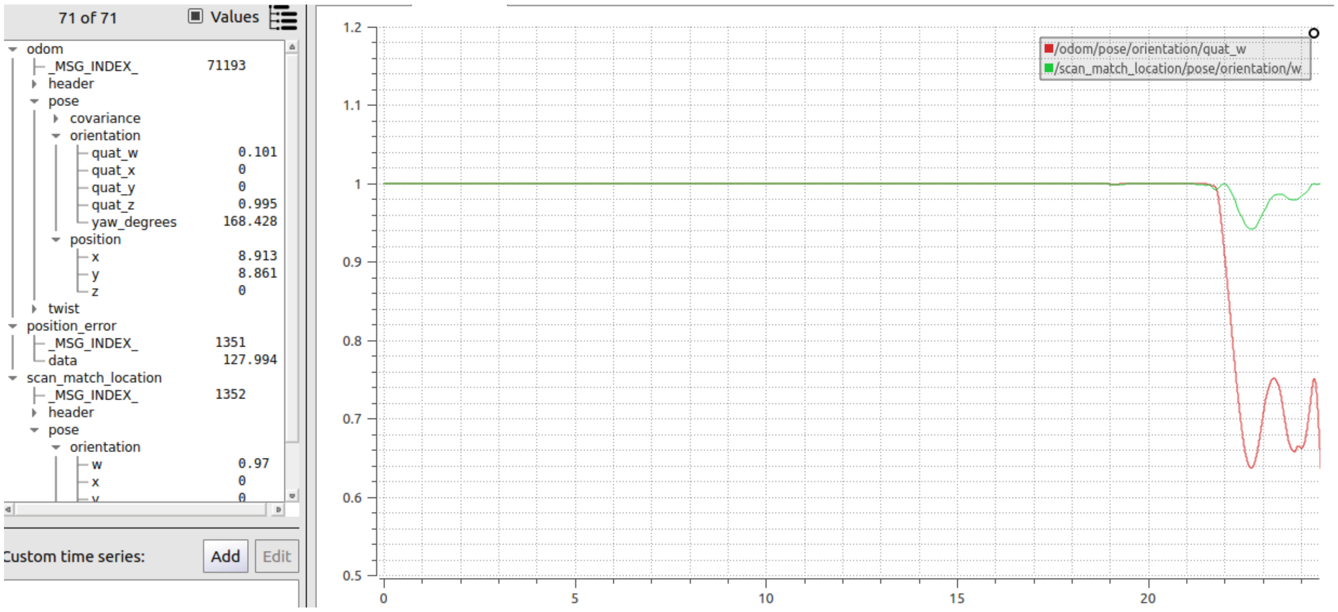


```
Filter...
69 of 69                          Values
odom
  _MSG_INDEX_                      58407
  header
  pose
    covariance
    orientation
      quat_w                           0.629
      quat_x                           0
      quat_y                           0
      quat_z                           0.777
      yaw_degrees                    102.004
    position
  twist
scan_match_location
  _MSG_INDEX_                      245
  header
  pose
    orientation
      w                                0.101
      x                                0
      y                                0
      z                                0.995
    position
      x                               -5.707
      y                               -8.327
      z                                0
Custom time series:          Add   Edit
```

plot
plot
plot
plot
/scan_match_location/pose/position/x
/odom/pose/position/x
/scan_match_location/pose/position/y
/odom/pose/position/y

With more iterations, the convergence will be slower but more accurate and so will be the results. We chose number_iter=1 for better performance due to the fact that any larger value causes latency issues between the car's true position and the converged transform.

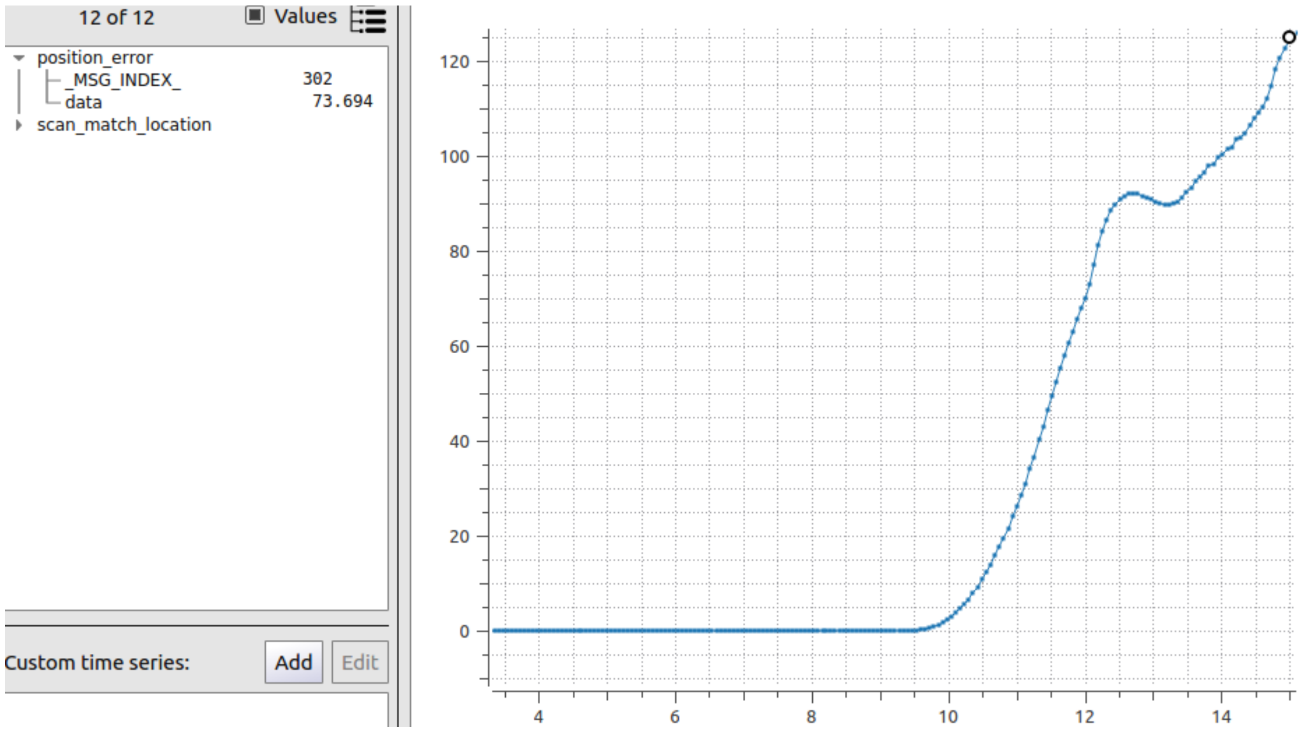
we also plot the w in quaternion:



```
71 of 71                          Values
odom
  _MSG_INDEX_                 71193
  header
  pose
    covariance
    orientation
      quat_w                    0.101
      quat_x                    0
      quat_y                    0
      quat_z                    0.995
      yaw_degrees             168.428
    position
      x                         8.913
      y                         8.861
      z                         0
  twist
position_error
  _MSG_INDEX_                  1351
  data                        127.994
scan_match_location
  _MSG_INDEX_                  1352
  header
  pose
    orientation
      w                         0.97
      x                         0
      y                         0
Custom time series:     Add   Edit
```

/odom/pose/orientation/quat_w
/scan_match_location/pose/orientation/w

To evaluate the performance, we add the analysis.py and plot the MSE with time. We set the MSE with the change of time (use $(\text{odom} - \text{match})^2$). Therefore, in the picture, we can see the error increased with time.



position_error
_MSG_INDEX_ 302
data 73.694
scan_match_location

In the figure above the car start moving at x = 10.

4. Conclusion

This PL-ICP method can work well in short distances and small rotations, but when we turn and change directions, the orientation will not be accurate and the errors accumulate. In this case, one iteration is enough for good performance because the impact of more computation iterations hurts more compared with and increase in accuracy.