

# 总体框架

cartographer大致可以分为三个部分，分别是传感器数据预处理、前端和后端，传感器预处理完成对点云进行带通滤波、体素滤波、运动畸变的去除操作，然后提供给前端。前端使用scan2map的方式进行扫描匹配，将当前点云与子图进行匹配，通过实时相关匹配算法和ceres非线性优化算法，计算出当前位姿，并根据运动变化量提取关键帧，在关键帧处对子图进行更新。后端将前端计算得到的关键帧以及子图位姿加入到位姿图中，对每个关键帧计算约束，包含子图内约束和子图间约束，在完成约束计算的关键帧达到一定数量之后，进行一次位姿图后端优化，优化位姿图中所有关键帧和子图的位姿，从而减小前端造成的累计误差并且完成回环检测，最终完成定位与建图功能。

## 前端

前端分为两个步骤，预测和更新。预测部分使用上一帧后验位姿，根据匀速运动模型计算出当前帧的先验位姿，不确定度增大，更新部分通过scan2map扫描匹配，计算得到当前帧的后验位姿，不确定度减小。当某一帧通过运动滤波后，选定这一帧为关键帧，根据关键帧的点云对子图进行更新，完成建图。最后将关键帧以及对应的点云添加到位姿图中。

## 前端预测

当不使用里程计数据和IMU数据时，预测部分根据匀速运动模型，根据上一帧的后验位姿、速度以及时间戳，计算得到当前的先验位姿。当使用IMU和里程计数据时，优先使用IMU数据的角速度预测先验位姿的姿态、优先使用里程计计算得到的速度计算先验位姿的位置

## 前端更新——扫描匹配

前端的更新步骤通过scan2map的方式，通过当前帧点云和子图的匹配，计算出当前的后验位姿。

### 大致流程

前端存在两种扫描匹配算法，分别是快速相关扫描匹配算法和ceres非线性最小二乘算法。快速相关扫描匹配算法是在先验位姿处生成一个搜索框，在搜索框内生成一系列经过一定平移和旋转的不同候选位姿以及对应的点云，通过暴力搜索的方式，计算出其中得分最高的候选位姿，作为ceres非线性优化扫描匹配算法的初值。ceres构建非线性最小二乘问题，参数块是当前位姿，残差块存在三种，分别是点云子图匹配残差、平移变化量残差、旋转变换量残差，经过ceres进行非线性优化，最终计算出当前帧的后验位姿。

### 两种算法的关系

ceres非线性优化扫描匹配算法所构造的代价函数经常是非凸函数，因此对参数块当前位姿的初值十分敏感，如果预测部分得到的先验位姿不好，可能导致非线性优化过程中陷入局部最小值，从而无法计算出最佳的后验位姿，所以可以使用快速相关扫描匹配算法，在先验位姿附近进行暴力搜索，找到一个匹

配程度较好的位姿，从而为ceres非线性优化扫描匹配算法提供更好的初值，这样就提高了非线性优化的效果。

## 快速相关扫描匹配器

根据选项参数生成搜索框，搜索框内生成一系列经过平移和旋转后的候选位姿及其对应的当前帧点云，进行暴力匹配，计算搜索框内每一个候选位姿的得分，得分越高，说明该候选位姿越好。最终将搜索框中得分最高的候选位姿作为计算结果，作为ceres非线性优化扫描匹配的初值。该得分与点云子图匹配程度、候选位姿的平移变化量、姿态变化量决定。计算候选位姿下的当前点云中每一点对应子图栅格地图中障碍概率的平均值，对于良好匹配的情况，当前点云恰好与子图栅格地图中障碍栅格重合，因此该平均值越高，匹配程度就越好，其次根据候选位姿与初始位姿在平移和旋转方面的变化量，对得分进行加权，变化量越大，得分越低。

## ceres非线性优化扫描匹配器

扫描匹配可以通过构建非线性最小二乘问题求解。该非线性最小二乘问题参数块是当前位姿，分别是xy方向上的位移，以及横摆角yaw。

存在三个残差块，分别是点云子图匹配残差块、位置变化量残差块、姿态变化量残差块。点云子图匹配残差块描述当前帧点云与子图的匹配程度，残差是当前位姿下点云中所有点在子图栅格地图中的对应的自由概率值，子图中的栅格地图保存了每个栅格的自由概率值，子图中障碍物的自由概率值为0，对于点云和子图完全匹配的情况，点云中每个点恰好对应子图中的障碍物，对应子图中的概率值恰好为0，所以残差越小，说明点云和子图匹配程度越高。位置变化量残差块、子图变化量残差块描述了更新步骤的可信程度，残差是扫描匹配的优化位姿与初值的变化量，使得计算出的更新位姿不会与初始位姿偏差过大。

对于点云子图扫描残差块的计算，需要将离散的2D栅格地图作为代价地图的一部分，但是位姿变化又是连续的，因此必须将离散的栅格地图进行插值，转换为一个连续的二元函数，才能计算对应的代价函数。

## 残差块权重系数调整

对于每个残差，cartographer都提供了对应的权重参数，描述了在求解非线性最小二乘问题的过程中，该残差所占的权重。

如果环境中几何特征不明显，匹配提供的残差会比较小，可以适当加大点云子图匹配残差权重，增加优化过程中点云和子图匹配的权重。如果更新步骤的扫描匹配比预测步骤的可信度低，可以适当加大位置和旋转权重，使得最终的优化位姿更接近于预测步骤得到的位姿。例如几何对称的环境扫描匹配可能出现误匹配，或者2DSLAM中出现上下坡这样的情况。另外可以根据已知的运动规律调节位移权重和旋转权重，例如，如果知道机器人在运动的过程中几乎不发生旋转，可以将旋转的权重适当增大，如果机器人经常急加速和急减速，可以适当减小位移权重。

## 前端子图更新

## 简介

根据运动滤波选定关键帧，也就是如果当前帧的位移、姿态、时间戳变化都超过对应的阈值之后，就选择这一帧为关键帧。点云中所有点在子图概率栅格地图中对应着障碍物，激光雷达到所有点云中的区域代表着自由区域，对概率栅格地图进行更新。

## 概率栅格地图

cartographer中是通过几率来完成概率栅格地图的更新的，几率是指某个栅格占据概率与栅格自由概率的比值，如果几率等于一，说明这个栅格占据和自由的概率各占一半，大于零小于一说明这个栅格自由的概率更大，大于一说明这个栅格占据的概率更大。

对于某次测量，得到的是这一栅格是自由还是占据的两种状态，在更新过程中，如果这个栅格之前没有测量过，那么直接赋予初始值。如果这个栅格得到过测量，存在一个先验几率，由贝叶斯定理可得，后验几率等于似然乘以先验几率，似然是由传感器本身决定的，传感器一旦确定，这个值就是固定的。在子图更新过程中，对贝叶斯等式两边取对数，如果某一栅格的测量是占据事件，那么后验几率对数等于占据似然对数加先验几率对数，如果某一栅格的测量是自由事件，那么后验几率对数等于自由似然对数加先验几率对数。

（公式推导可参考笔记本）

在cartographer中，由于传感器一旦确定，似然就是固定值，因此先验几率和后验几率存在一一对应的映射关系，因此为了减少计算了，首先计算所有先验对应的后验，保存为一个更新表，在实际更新过程中，根据这个更新表就可以找到占据事件或更新事件下，先验几率对应的后验几率，节省了计算资源。

## 后端

### 回环检测——计算子图间约束

后端的位姿图每接收一个关键帧，会计算该关键帧到它附近一定距离内的子图的约束位姿，为了计算该约束位姿，需要该关键帧点云与子图进行扫描匹配。回环检测部分的扫描匹配使用了两种算法，分别是快速相关扫描匹配器和ceres扫描匹配器。最终获得该关键帧与子图的约束位姿，等待后端优化。

### 快速相关扫描匹配算法

因为回环检测时初始位姿的误差通常比较大，非线性优化对初值也比较敏感，因此需要使用相关方法得到进行匹配，将匹配的结果作为非线性优化的初值，才能成功进行扫描匹配。与前端的相关方法相比，回环检测时由于初始位姿误差较大，需要创建较大的搜索框，会产生很多的候选位姿，所以无法向前端那样进行暴力搜索。快速相关方法使用了分支定界的方法对计算进行加速，从而实现了计算的实时性，首先根据原始的栅格地图，生成一系列不同分辨率的地图，低分辨率地图的栅格中保存的是原始栅格地图对应栅格中的最大值，因此对元素栅格地图进行了简化的同时保留了栅格地图中的最大值信息，首先在最低分辨率地图中生成一系列经过旋转和平移的点云，计算点云投影至低分辨率栅格地图中对应占据

概率的平均值，该平均值越大，说明匹配程度越高，在完成匹配后使用分支定界方法，最终找到原始栅格地图中平均值最高的点云，它就是当前搜索框内匹配程度最高的位姿。

## 分支定界算法

分支定界利用深度优先搜索和剪枝的算法思想，所谓定界是指只对当前分辨率地图中得分高于下界的位姿进行分支，避免不必要的计算，所谓分支是指当前分辨率地图中的某一位姿，在它分辨率更高的地图中对它附近进行匹配，获得分辨率更高地图中的匹配结果，由粗到精地寻找得分最高的位姿。由于分支定界的目标是找到最大值，因此在分辨率更高的地图中进行扫描匹配时，不需要对得分低于上一层地图匹配的最大值的点云进行匹配，因此在每一层分支定界中下界会随着上一层的最大得分提高，因此在由粗到精的过程中避免了很多不必要的计算，实现了相关方法计算的实时性。

## ceres扫描匹配器算法

在获得快速相关扫描匹配结果后，将它作为初值进行ceres扫描匹配，计算更加准确的结果，这一部分的算法与前端是相同的

## 后端优化

后端分为约束计算和位姿图非线性优化两部分。新添加到位姿图的关键帧计算它在位姿图中的约束，当完成约束计算的关键帧个数每达到一定数量，根据当前位姿图进行一次的非线性优化。

位姿图由节点和边组成，位姿图中的节点代表位姿，分为是关键帧位姿和子图位姿，边代表两节点之间的约束，是两节点之间的相对位姿，分为子图内约束和子图间约束。

### 约束计算

前端产生关键帧之后，就会将该关键帧位姿和它所属的子图位姿添加到位姿图中，然后分别添加该关键帧的子图内约束位姿和子图间约束位姿，子图内约束是指由前端扫描匹配得到的该关键帧与它所属子图之间的相对位姿，负责子图和其中关键帧之间的一致性；子图间约束位姿是指该关键帧附近一定范围内的子图经过后端扫描匹配得到的关键帧与该子图的相对位姿，负责回环优化。当完成约束计算的关键帧个数每达到一定数量，就进行一次位姿图的非线性优化。

### 位姿图非线性优化

后端位姿图非线性优化，通过构建非线性最小二乘问题优化位姿图，减小前端的累计误差，同时完成回环检测。

首先进行非线性最小二乘问题的建模。参数块是当前位姿图中所有子图和关键帧的位姿，残差是约束位姿与相对位姿之间的差值，对于不使用GPS和里程计的情况下，首先添加当前位姿图中所有子图间约束位姿和子图内约束位姿残差，代价函数是约束位姿和该约束对应子图和关键帧之间相对位姿的差值；其次添加当前位姿图中按时间顺序的两两关键帧之间的前端位姿残差，代价函数是前端计算的两关键帧之

间的相对位姿和后端优化得到的相对位姿之间的差值。在完成非线性最小二乘问题的建模之后，进行非线性优化求解，根据优化后的结果更新位姿图中各个关键帧和子图的位置。

## 调参

首先调整带通滤波的最短距离和最远距离，由于在转向是激光雷达受到了一部分车体的遮挡，因此需要排除距离激光雷达过近的点，由于激光雷达本身的精度与距离有关，铲运车运行的场景比较狭小，因此设定最远距离为三十米。

### 前端调参

对于前端来说，需要保证求解位姿的准确性和健壮性。通常是需要设置合适的残差权重，前端的实时相关匹配器和ceres非线性优化匹配器包含扫描匹配残差、平移变化量残差、旋转变换量残差，其分别存在不同的权重。由于加速度比较大，铲运车在加速过程中有时前端会出现无法推算位姿的情况，其原因是急加速过程中匀速运动模型预测得到的先验位姿误差较大，ceres非线性优化器对初始值比较敏感，因此在优化过程中容易陷入局部最小值，从而无法计算出正确的后验位姿。所以我适当加大了搜索框的长度，并且适当减小了平移变化量残差权重，使得实时相关匹配器可以找到距离先验位姿相对较远的最佳匹配位姿，然后在进行非线性优化，通过合适的参数调整，前端稳定性大大提高。

反之，如果受到环境或者传感器本身的影响，前端里程计相比轮式里程计不准确，可以适当增大平移和旋转权重，使得由轮式里程计得到的更新的先验位姿权重更大。

### 后端调参

对于后端来说，需要保证优化的实时性和健壮性。首先要控制位姿图中关键帧参与约束计算的数量，后端是通过一定的采样频率来控制位姿图中参与优化计算的关键帧，采样率过高，会大大增加计算负担降低实时性。对于回环约束的匹配，可以根据建图时的累计误差，选定回环约束的搜索框大小、回环约束范围、残差权重，如果进行回环的时候误差较大，可以适当增大回环约束范围、搜索框大小。

后端非线性优化的频率是通过完成约束计算的关键帧数量决定的，位姿图中每计算完成这些数量的关键帧，就对位姿图进行一次非线性优化，为了能够及时消除位姿图中的误差，可以适当降低这个参数，提高位姿图非线性优化的计算频率。

后端优化时总和考虑了许多残差，包括约束残差，前端里程计残差、轮式里程计残差、GPS残差，每个残差都由相对应的权重系数，如果某残差比较准确，则可以适当增加对应的残差权重，提高它在后端优化的权重。

### 纯定位调参

定位与建图的调参策略不同，定位需要较高的实时性，因此需要大大提高位姿图优化的频率，能够及时消除定位误差。由于定位实际上是当前位姿与以及建立完成的子图进行扫描匹配，因此会产生大量子图间约束，所以要适当减小关键帧计算约束的采样频率，提高计算的实时性。

## cartographer调参笔记

## 建图前端相关

`TRAJECTORY_BUILDER_2D.min_range` 表示最近点云距离，小于该距离的点云会被过滤。当车身会部分遮挡激光雷达且无法调整时，通过设置该参数去除车身遮挡对建图定位的影响

`TRAJECTORY_BUILDER_2D.max_range` 表示最远点云距离，在最远和最远距离之间的点云用于扫描匹配。通过设置该参数可调整子图的大小，为提高扫描匹配的效果，应尽量增大最远点云距离，但应考虑激光雷达精度与建图的精度，该参数不应设置过大，避免最远处的点云误差对建图带来影响，通常场景越大，该参数应相应增大

`TRAJECTORY_BUILDER_2D.missing_data_ray_length` 超过最远距离的激光点的扫描距离，可设置为最远距离的三分之二

`TRAJECTORY_BUILDER_2D.num_accumulated_range_data` 累计点云数量，当累计达到该数目的点云之后，在进行本地SLAM，需要与`num_laser_scans`和`num_subdivisions_per_laser_scan`相符合、

`TRAJECTORY_BUILDER_2D.use_online_correlative_scan_matching` 是否使用实时相关扫描匹配器进行本地SLAM，当里程计和IMU数据不准确或只使用激光雷达进行SLAM时，应该开启相关扫描匹配器。由于ceres扫描匹配器对初值较为敏感，当无法提供较为准确的预测位姿时，ceres扫描匹配器难以收敛，表现为匹配点云经常无法更新位姿或发送误匹配导致较大漂移，当开启相关扫描匹配器后，其会使用相关方法在预测位姿附近产生指定大小的搜索框，遍历该搜索框所有的xy0位姿下的点云，进行scan-to-map扫描匹配，根据匹配程度、平移量和旋转量找到的分最高的位姿，将其作为初值提供给ceres扫描匹配器，使其获得更精确的初值，而代价是占用额外的计算资源，影响本地SLAM的实时性。

`TRAJECTORY_BUILDER_2D.real_time_correlative_scan_matcher.linear_search_window` 实时扫描匹配器的线性搜索框大小。实时相关扫描匹配器会以预测位姿为中心，以该参数的两倍为边长的正方形内遍历所有栅格。由于实时相关扫描匹配器未使用分枝定界加速，并且本地SLAM对实时性要求较高，因此该参数不宜设置过大。

`TRAJECTORY_BUILDER_2D.real_time_correlative_scan_matcher.angular_search_window` 实时扫描匹配器的角度搜索框的大小。实时相关扫描匹配器会在线性搜索框内所有的栅格，遍历所有正负该参数内的角度，与线性搜索框大小一样，该参数不宜设置过大。

`TRAJECTORY_BUILDER_2D.real_time_correlative_scan_matcher.translation_delta_cost_weight` 平移量权重，该权重越大，平移量越小的位姿得分越高，最终输出的位姿位置与预测位姿位置越近。该参数过大时，会导致更新位姿平移量过小，即无法更新位姿，该参数过小，会导致轨迹不够平滑，经常发生误匹配。当环境特征充足，激光点云可以获得较好的更新位姿时，该参数可适当减小，（未验证：当使用里程计计算预测位姿且里程计精度较高而环境特征不充足时，应适当增大该参数）

`TRAJECTORY_BUILDER_2D.real_time_correlative_scan_matcher.rotation_delta_cost_weight` 姿态旋转权重，该权重越大，姿态旋转越小的位姿得分越高，最终输出的位姿姿态与预测位姿姿态越近。该

参数过小时，旋转时会产生误差。

TRAJECTORY\_BUILDER\_2D.ceres\_scan\_matcher.occupied\_space\_weight ceres扫描匹配器scan-to-map匹配权重，该权重越大，匹配点云与子图匹配误差影响越大，即与子图越匹配的位姿更有可能成为更新位姿，当环境特征较为明显时，可适当增大该参数，提升位姿更新效果，为避免误匹配，该参数不宜设置过大。（未验证：当使用里程计时，当环境特征不明显导致scan-to-map无法准确匹配时，可适当减小该参数）

TRAJECTORY\_BUILDER\_2D.ceres\_scan\_matcher.translation\_weight 扫描匹配器平移权重，该权重越大，平移量带来的影响越大，与初值越近的位姿更有可能成为更新位姿。该参数过大时，会导致更新位姿平移量过小，即无法更新位姿，该参数过小，会导致轨迹不够平滑，经常发生误匹配。当环境特征充足，激光点云可以获得较好的更新位姿时，该参数可适当减小，（未验证：当使用里程计计算预测位姿且里程计精度较高而环境特征不充足时，应适当增大该参数）

疑问：ceres\_scan\_matcher.translation\_weight和

real\_time\_correlative\_scan\_matcher.translation\_delta\_cost\_weight 应当优先调整哪一个？

TRAJECTORY\_BUILDER\_2D.ceres\_scan\_matcher.rotation\_weight 描匹配器姿态旋转权重。该权重越大，姿态旋转越小的位姿得分越高，最终输出的位姿姿态与预测位姿姿态越近。该参数过小时，旋转时会产生误差。

TRAJECTORY\_BUILDER\_2D.pose\_extrapolator.use\_imu\_based 位姿推算器是否使用IMU数据。通过使用IMU数据的角速度和线加速度预测位姿的3D姿态，通过里程计的位姿计算的线速度角速度来预测位姿的位置。不是用IMU时，不能预测3D姿态，只能预测2D姿态

TRAJECTORY\_BUILDER\_2D.pose\_extrapolator.constant\_velocity.pose\_queue\_duration 位姿推算器位姿队列时间长度，该参数越大，位姿推算器队列保存的更新位姿越多。在没有里程计和IMU的情况下，位姿推算器保存过去一段时间的位姿，通过队列中最新和最旧的位姿以及时间差计算平均速度，将其作为车辆当前的瞬时速度，预测下一时刻的车身位姿。当经常发生误匹配导致速度计算异常时，可适当稍微增大该参数

TRAJECTORY\_BUILDER\_2D.pose\_extrapolator.constant\_velocity.imu\_gravity\_time\_constant 位姿推算器位重力加速度更新参数，该参数越大，IMU线加速度对重力加速度更新的权重越小（未验证：当IMU噪声过大时，可适当增大该参数）

TRAJECTORY\_BUILDER\_2D.motion\_filter.max\_time\_seconds

TRAJECTORY\_BUILDER\_2D.motion\_filter.max\_distance\_meters

TRAJECTORY\_BUILDER\_2D.motion\_filter.max\_angle\_radians

轨迹节点滤波器参数，只有通过该滤波器的更新位姿才会成为轨迹节点，对子图进行更新。根据累计激光点云的频率，会计算出一系列的更新位姿，每个更新位姿都包含其对应的匹配点云，然而不是所有的

更新位姿都对子图进行更新，只有轨迹节点才会对子图进行更新，轨迹节点即对子图进行更新且参与全局SLAM的更新位姿，当上述其中一个条件不满足时，该更新位姿被选取为轨迹节点。

TRAJECTORY\_BUILDER\_2D.submaps.num\_range\_data 新建子图时的匹配点云数量

TRAJECTORY\_BUILDER\_2D.pose\_extrapolator.constant\_velocity.imu\_gravity\_time\_constant = 10  
IMU线加速度更新重力加速度系数，该系数越大，IMU数据更新权重越小。实测该参数过小时，IMU线加速度噪声或车身加速会对重力加速度方向带来较大的影响，显示为车身加速时点云会产生摆动

建图后端相关

TRAJECTORY\_BUILDER\_2D.submaps.num\_range\_data

后端优化的效果与子图的大小有很大的关系，子图过小，单个子图的特征较为不明显，对扫描匹配精度存在负面影响，且增大计算量浪费计算资源，还会对前端产生影响；子图过大，前端误差会累计在单个子图中，使得后端无法消除此误差，综上所述，应根据速度、场景大小、匹配最大距离、节点采样率综合判断

POSE\_GRAPH.constraint\_builder.max\_constraint\_distance

回环约束中轨迹节点与子图的最大距离，小于该距离的回环约束不参与计算。回环约束在进行回环检测即车辆建图后返回到之前的位置的时候再进行回环约束的计算，应保证回环约束不在轨迹节点之前的几个子图内进行计算，避免浪费计算资源

POSE\_GRAPH.constraint\_builder.sampling\_ratio、

节点中参与约束计算的采样率。过高的采样率会浪费系统资源，降低实时性。

POSE\_GRAPH.constraint\_builder.min\_score

计算回环约束时进行相关扫描匹配时的最低得分，舍弃低于该得分的某节点扫描匹配得到的最佳约束位姿。提高该参数，可舍弃一些匹配较差的约束位姿，若进行回环检测时，经常发生误匹配，可根据日志打印的得分直方图适当提高该参数

POSE\_GRAPH.constraint\_builder.fast\_correlative\_scan\_matcher.linear\_search\_window、

POSE\_GRAPH.constraint\_builder.fast\_correlative\_scan\_matcher.angular\_search\_window

相关扫描匹配器的搜索框大小参数。回环约束是在轨迹节点处由该参数产生的搜索框内进行扫描匹配获得的，因此建图进行回环检测时前端累计的误差较小，可适当减小该参数，避免产生误匹配导致的错误的回环约束

POSE\_GRAPH.matcher\_translation\_weight

POSE\_GRAPH.matcher\_rotation\_weight

子图内约束残差平移旋转权重。子图内约束用于保持全局优化轨迹相邻子图的连续性，该约束残差权重越大，进行优化时该残差的消除效果就越大，对应子图连续性的较大保持程度。



`POSE_GRAPH.constraint_builder.loop_closure_translation_weight`

`POSE_GRAPH.constraint_builder.loop_closure_rotation_weight`

子图间约束即回环约束残差平移旋转权重。回环约束用于进行回环检测，由当前的轨迹节点在之前的子图内进行相关扫描匹配，获得的批评位姿作为约束位姿，而当前轨迹的位姿与约束位姿之间的差值为残差。该参数越大，约束位姿的权重就越大

`POSE_GRAPH.optimization_problem.huber_scale`

回环约束残差的 huber损失函数系数。当不可避免地产生错误的回环约束时（通常残差较大），可通过该参数削弱错误回环约束的影响。

由于相关扫描匹配器误匹配导致的错误回环约束。图中黄线代表约束位姿，由该节点在该子图的相关扫描匹配器经过扫描匹配获得的相对匹配位姿，即被认定为该轨迹节点的“真实”位姿，图中的蓝线代表约束位姿残差，即不可信的当前轨迹节点位姿与可信的约束位姿之间的差值，即后端优化的残差

`POSE_GRAPH.optimization_problem.local_slam_pose_translation_weight`

`POSE_GRAPH.optimization_problem.local_slam_pose_rotation_weight`

本地SLAM输出轨迹的优化权重。若环境特征明显且激光雷达精度较高，可适当增大该参数，若环境特征若例如光滑且较长的走廊，可适当减小该参数。

`POSE_GRAPH.optimization_problem.odometry_translation_weight`

`POSE_GRAPH.optimization_problem.odometry_rotation_weight`

里程计输出轨迹的优化权重。若里程计精度较高且本地SLAM输出位姿精度较低，可适当提高该参数，反之可适当降低该参数。

`POSE_GRAPH.optimization_problem.fixed_frame_pose_translation_weight`

`POSE_GRAPH.optimization_problem.fixed_frame_pose_rotation_weight`

GPS定位输出位姿优化权重。若GPS信号良好且精度较高，可尽量提高该参数。

定位相关

前端：

为保证前端稳定性，可以在保证激光雷达精度的情况下，尽量提高参数

`TRAJECTORY_BUILDER_2D.max_range`，并减小运动滤波器的参数，以增加轨迹节点的个数。

为保证定位时建立正确的约束。首先根据轨迹节点的个数设定

`POSE_GRAPH.constraint_builder.sampling_ratio`，根据定位时车辆的通行区域，设定最大约束距离

`POSE_GRAPH.constraint_builder.max_constraint_distance`，避免参与计算的约束数量过多降低实时性，并观察定位时出现的残差，设定搜索框参数观察得分直方图，设定最低得分

`POSE_GRAPH.constraint_builder.min_score`，过滤不良匹配，

## cartographer 日志信息

### 前端相关

scan rate: 10.00 Hz 1.00e-01 s +/- 5.23e-04 s (pulsed at 100.60% real time)

记录接收各个传感器数据的统计数据，分别记录传感器数据接受频率、时间间隔均值、方差、传感器数据接收的实时性

Remaining work items in queue: 0

记录轨迹节点个数达到参赛规定的条件后进行全局SLAM时，工作队列中剩余的工作函数个数。工作函数负责接收传感器数据，计算轨迹节点的约束

Node (0, 60) with 201 points on submap (0, 0) differs by translation 0.01 rotation 0.000 with score 71.7%.

当该节点和该子图的约束计算完毕后，分别记录轨迹节点ID、该节点匹配点云中激光点的个数、子图ID、匹配位姿与初始位姿的平移量和旋转量、匹配得分

4 computations resulted in 4 additional constraints.

记录在进行全局SLAM之前，本轮全局SLAM参与计算的约束个数（即通过了约束距离条件和定比例滤波器的约束）和存在计算结果的约束个数（即最终匹配得分高于参数中制定的最低得分的约束）

Score histogram:

Count: 90 Min: 0.551582 Max: 0.801629 Mean: 0.684009

[0.551582, 0.576587) ## Count: 9 (10%) Total: 9 (10%)

[0.576587, 0.601591) # Count: 4 (4.44444%) Total: 13 (14.4444%)

[0.601591, 0.626596) ## Count: 7 (7.77778%) Total: 20 (22.2222%)

[0.626596, 0.651601) ### Count: 8 (8.88889%) Total: 28 (31.1111%)

[0.651601, 0.676605) #### Count: 14 (15.5556%) Total: 42 (46.6667%)

[0.676605, 0.701610) ### Count: 8 (8.88889%) Total: 50 (55.5556%)

[0.701610, 0.726615) ## Count: 7 (7.77778%) Total: 57 (63.3333%)

[0.726615, 0.751620) #### Count: 14 (15.5556%) Total: 71 (78.8889%)

[0.751620, 0.776624) #### Count: 12 (13.3333%) Total: 83 (92.2222%)

[0.776624, 0.801629] ## Count: 7 (7.77778%) Total: 90 (100%)

记录在进行全局SLAM之前，本轮全局SLAM所有约束的得分统计直方图。

第一行显示本轮全局SLAM所有新的约束的个数、最低得分、最高得分、平均得分之后为直方统计图

Motion filter reduced the number of nodes to 34.2%.

运动滤波后，轨迹节点占所以更新位姿的个数

Inserted submap (0, 9).

新建一个子图，子图ID为轨迹ID和子图在该轨迹下的序号