学习动机

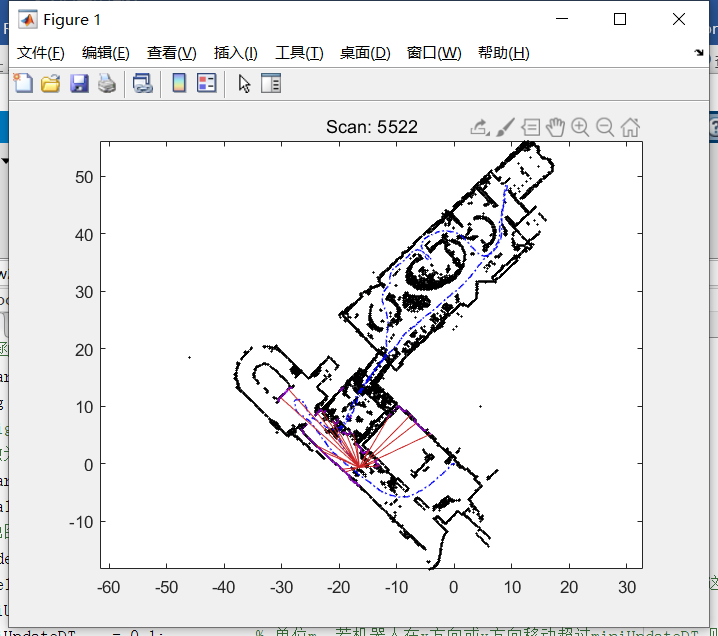
|  |  |
| --- | --- |
| 正向 | 反向 |
| 熟悉SLAM整套发展流程 | 没有项目空谈理论 |
| 对各种的库的使用程度得到展现 | 无法和国内研究生一起竞争 |
| 多种策略结合 证明看过多种框架 | 比没有项目刚刚想入门的人强很多 |
| C++ 熟悉程度对于SLAM 开发基本够用 |  |
| SLAM 2d 激光slam的基础知识基本可以过 |  |
| 多传感器融合经验 |  |
| 熟悉Cartographer 的代码 可以上手进行修改 |  |

主要就是展示自己、增强竞争力

🡪Matlab 2D SLAM理清程序逻辑、调试完成

|  |  |  |
| --- | --- | --- |
| 1. 参数设置 | * 搜索窗添加边界的尺寸 * 像素尺寸 一个像素代表的长度 * 添加的关键帧的参数(运动滤波器) | * CSM参数设置 搜索分辨率 FastResolution * 闭环参数设置 搜索分辨率 BruteResolution |
| 1. 整体初始化 | * 全局map初始化 | |
| 1. 扫描帧的收集和预处理 | * 提取单个帧 * 只取一定range范围内的扫描点 | |
| 1. CSM | * 提取scan周围可能的局部地图 * 局部地图生成栅格地图 | * 恒定速度模型提供预测初始值 * Hill-Climbing 搜索算法 |
| 1. 关键帧处理 | * 判断关键帧 * 寻找和上一个keyscan的新点，然后添加keyscan，添加connection用于之后的图优化 | |
| 1. 回环 | * 检查是否进行回环 >10 个关键帧 并且travel距离需要大于10m * 确定回环检测进行的关键帧的范围 轨迹长度小于50 并且找到一个与当前帧距离较近的关键帧 * 非常暴力 –r –x –y 的CSM基本这个回环是费的 没有办法使用 | |

效果图



🡪在2D SLAM 加入需要加入的内容:体素滤波、分支定界CSM、两次三双差值、submaps的搜索结构、回环

|  |  |  |
| --- | --- | --- |
| 1. 参数设置 | * 搜索窗添加边界的尺寸 * 像素尺寸 一个像素代表的长度 * 添加的关键帧的参数(运动滤波器) * **Submaps的参数设置 ☑** | * CSM参数设置 搜索分辨率 FastResolution * 闭环参数设置 搜索分辨率 BruteResolution |
| 1. 整体初始化 | * 全局map初始化 **keyframe 换成cartographer的trajectyory\_node submaps ☑** | |
| 1. 扫描帧的收集和预处理 | * 提取单个帧 * 只取一定range范围内的扫描点 * **体素滤波 留下自适应滤波之后做 ☑** | |
| 1. CSM | * **Cartographer submap 匹配 ☑** * 提取scan周围可能的局部地图 * 局部地图生成栅格地图(HITS)地图 确定很明显只要扫到了就确定存在的地图，在很多现实场景中没有概率分布的概念，一切都是一个确定的值，一旦有动态环境出现，全部完蛋。 相当于没有利用好每一个扫描数据，只利用新数据 * **换成栅格地图(odds)** | * 恒定速度模型提供预测初始值 * Hill-Climbing 搜索算法在局部情况下要快得多 **–r-x-y 换成r 层**   **☑或者多分辨率**   * 精细化CSM **换成优化匹配** |
| 1. 关键帧处理 | * 判断关键帧 * 寻找和上一个keyscan的新点，然后添加keyscan，添加connection用于之后的图优化 **阅读Carto 的submap内容在全局地图中进行配置 ☑** | |
| 1. 回环 | * 检查是否进行回环 >10 个关键帧 并且travel距离需要大于10m **换成cartographer 的submap 插入结束进行回环的方式** * 确定回环检测进行的关键帧的范围 轨迹长度小于50 并且找到一个与当前帧距离较近的关键帧 * 非常暴力 –r –x –y 的CSM基本这个回环是费时的 没有办法使用 **分支定界的进行回环检测** | |

具体实现(Cartographer)

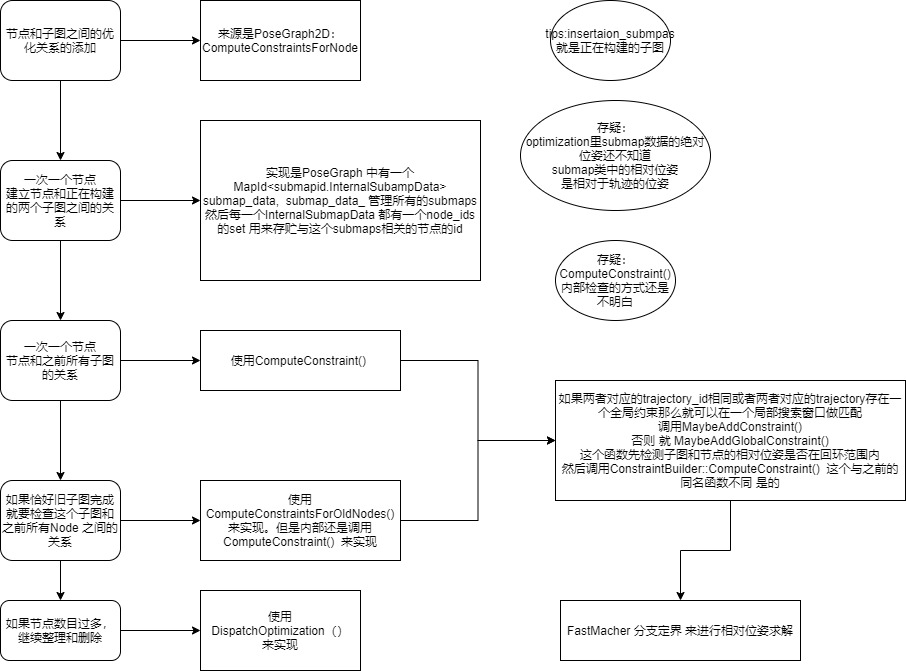
1. 体素滤波

|  |  |
| --- | --- |
| Cartographer 中使用Voxel Filter中的Filter()函数进行滤波：  使用一个voxel\_set\_ 的unordered\_set来存放voxel。  Results 是第一个近似成Voxel 的值  eg. (2.1,2.2)(2.05,2.1) 都近似成(2,2) results 用(2.1,2.2)  .second返回一个pair 其中包含一个迭代器 一个bool来表示是否插入  ☑  Matlab中使用unique 来实现 | TimedPointCloud VoxelFilter::Filter(const TimedPointCloud& timed\_point\_cloud) {  TimedPointCloud results;  for (const Eigen::Vector4f& point : timed\_point\_cloud) {  auto it\_inserted =  voxel\_set\_.insert(IndexToKey(GetCellIndex(point.head<3>())));  if (it\_inserted.second) {  results.push\_back(point);  }  }  return results;  } |

1. Submaps

|  |  |
| --- | --- |
| * Submaps 双子图结构 两个子图之间50% 重合 ☑ | |
| * 旧子图用于匹配，新子图保持构建 ☑ 看代码确定过了   在ActiveSubmaps2D AddSubmap有体现旧子图在前 新子图在后 | |
| * 每一个submaps内含概率栅格图，每一次插入帧就更新一次概率栅格图，目前先更新hits 图。 ☑ | |
| * 为了防止submaps内部帧过于密集加入motionFilter ☑ | |
| * 旧子图构建结束之后，进行回环方面的考虑 不如说是每一个节点插入之后，以一个节点为单位来进行回环方面的考虑 | |
| * Submaps 自身有pose ☑ | |
| * 理清submaps 和节点的关系☑ | |
| * 后端检测约束和submaps 的关系 ☑ 出处：   🡪GlobalTrajectoryBuilder：：AddSensorData( RangeData & )🡪AddNode()🡪ComputeConstraintsForNode()&ComputeConstraintforOldNodes🡪 ComputeConstraint()🡪MayAddConstraint()提供submap id 和node id 看是否有约束关系🡪MatchFullSubmap() & Match() | |
| **Submaps结构使用结果** | |
| 修改前效果 | 原因分析(猜测) |
| 1. 速度上相较于之前, (每一个扫描帧都在全地图点搜索附近点来进行匹配的方法)，没有太大提升。 *真正让整个算法加速的是预处理时进行的体素滤波* | 1.1 搜索地图点只是一个O(N)的算法，运动Kd-tree之后可以达到O(logN)的算法速度。所以在搜索地图点上，花费的时间并不多。Submaps 的优势更多的是在后端约束时，不用以Node 为单位来建立优化的约束。因为Matlab版本暂时还没有写一个大规模的后端优化，效果无法体现。 |
| 1. Submaps 结构相对于之前的结构，在近距离多次重复的轨迹上，有很大的不一致性。精度没有问题 | 2.1 构建的两个submaps 更新不及时导致的，因为每一次衡量一个是否添加keyscan都是在对pose 的距离进行评估，而不是对轨迹长度进行评估。这将直接导致2 的情况下，submaps无法及时进行更新，在匹配方面精度出现问题  **不是这个问题** |
| 2.2 Hill-Climbing的求解局部性的影响，实验Hill\_Climbing的局部性和这个局部性对于地图的敏感度。发现还是很敏感的，很容易达到一个局部最优的解。 把原版的局部地图(认为这个地图更向最优解集中)所得出的解放进submap体系中所得出的局部地图中进行打分，发现这个解会比从submap局部地图中得到的解更优。  AddKeyscan 因为用hits过滤，只添加新的坐标点，一旦匹配有一帧错误加入大量错误点，就容易出问题。 |
| 1. Submaps 结构在匹配时，如果内点不足需要扩大submaps搜索范围，集合多个submap进行匹配时，会需要大量的时间 | 3.1 这个是由keyscan 的结构决定，因为每一个keyscan 只保存了新的点，一旦场景相似或重复，会出现keyscan点数不足的问题。 |
| 修改策略 | 修改效果 |
| 1.1.1暂不处理，等待后端优化 |  |
| 2.1.1 修改Addkeyscan 的策略，增加一个轨迹评估和经过多少个扫描帧就增加一个关键帧 | 结果没有任何改善，Points 30000个左右。 |
| 2.2.1 把submap设置成一个只是用于后端图成像和便于优化的一个结构，而前端的估计还是使用原版 | 因为这种改变，在精度上已经没有差别了，也是一种警示吧，为什么cartographer CSM之后要加subpixel的Ceres 优化。如果仅仅是submaps 结构直接进行hill-Climbing 局部最优解，累计的误差会让系统出问题。原因hill-climbing 对于局部地图的敏感性太高了。 |
| 3.1.1 还是需要建立一个和orb一样的共视图的结构，加强submaps之间联系 |  |

后端检测约束和submaps 的关系



1. CSM submaps匹配

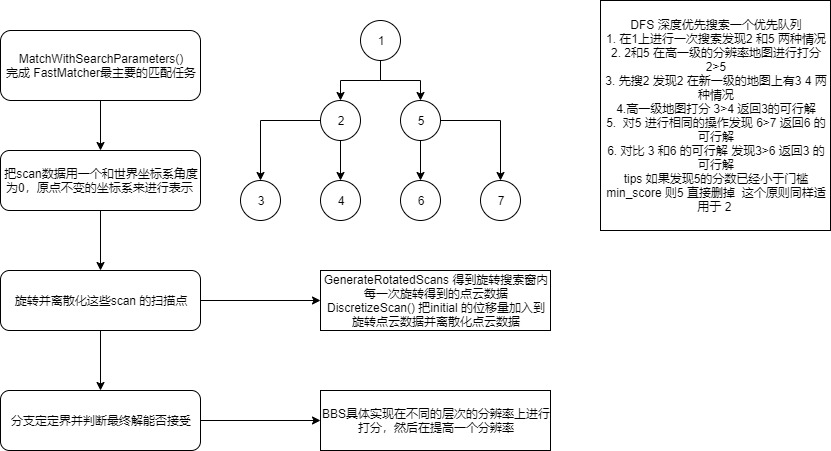
CSM原匹配方式 HITS 还没有上ODDS

|  |  |  |
| --- | --- | --- |
| 失败原因讨论 | 修改方式 | 得到的结果 |
| 猜想1：Submaps中存储的关键帧信息有问题，因为每一个keyscan中的关于扫描点的数据只是存储的新的点，如keyscan 非常密集，每个keyscan存储的点会很少  验证: 在进行子图匹配时，显式子图内点的数量可以很清楚的看出来 | 1.1.1 改变：keyscan变稀疏 submaps的存储量加大 | 1.1.2 结果：前期已经基本没有问题了，但是一旦有了重复的点submaps内存储点的数量会急剧下降🡪 直接导致匹配有问题 |
| 1.2.1改变: 我们在相信前期准确建图的情况下，在搜索内点过少时，根据submaps 的pose 寻找附近的submaps 一起进行匹配 | 1.2.2 结果：完全OK 可以使用了 |
| **后期进一步整改** | 整改方式 | 整改结果 |
| 问题1：一旦有大量的重复点出现的时候这个子图的搜素需要花费时间，要找一个方式，将搜索的时间消耗降低 | 猜想：1. 使用类似ORB 的共视图的形式 |  |

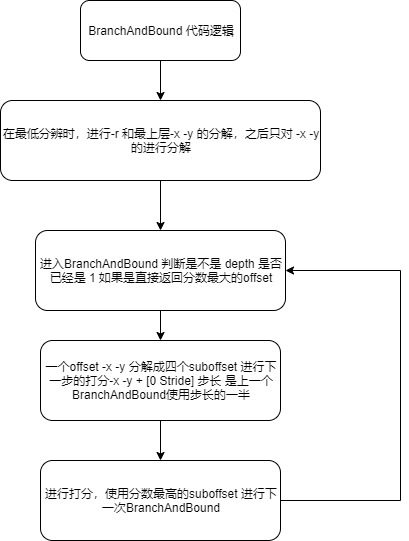
1. 回环FastMatcher研究

|  |  |
| --- | --- |
| 1. 理清分支定界的逻辑 ☑ | |
| 1. 理清多分辨率地图的构建 ☑ 基本确定用图像滤波来降低分辨率 和论文M3MRSM 有出入 但是核心思想 Equation5 in M3MRSM 是一致的 | |
| 1. 寻找每一步linear\_bound x y 的 min 和max ☑ | |
| 1. BBS模块设计 | * 地图的分辨率按照BBS的层数来进行模糊 * 单次行进的步长和BBS的层数有关 * 打分模块 * 递归的设计 * **优化：原来的只有固定的mini\_score 来进行过滤，可以考虑用一个全局变量试试更新最大的score 用这个score 当做门槛。** * **考虑比较高的分辨率的时候，使用Hill-Climbing算法来加速算法** |
| 1. 整体测试 | |
| 1. 设置图优化 约束存储的机制 | |

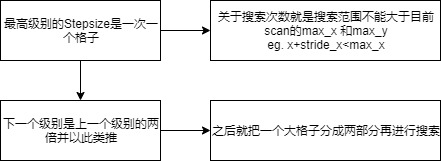
分支定界的逻辑



**TIPS： 分支定界FastScanMatcher之后 加一个CeresMatcher 精细化**



搜索范围逻辑



🡪Matlab 往C++ 转换，进行一次重构只处理激光数据，加入Ceres优化和线程池还有OpenCV点云显示

1. Ceres 学习

整体流程

|  |  |
| --- | --- |
| 1. Construct a Cost Function | The important thing to note here is that **operator()** is a templated method, which assumes that all its inputs and outputs are of some **type T**.  The use of templating here allows Ceres to call **CostFunctor::operator<T>()**, with **T=double** when just the value of the residual is needed, and with a special type**T=Jet** when the Jacobians are needed. |
| 1. 创建Problem | Problem problem; |
| 1. 计算Derivative并加入Probelm | Problem problem;  problem.AddResidualBlock(new AutoDiffCostFunction<F1, 1, 1, 1>(new F1), NULL, &x1, &x3);  <F1,1,1,1> 第一个数字是输出的维度，剩下两个是优化的维度(x1 和 x3)，F1 是优化方程 NULL 是核函数 |
| 1. Options 的配置 | Solver::Options options;  options.max\_num\_iterations = 100;  options.linear\_solver\_type = ceres::DENSE\_QR;  options.minimizer\_progress\_to\_stdout = true; |
| 1. 进行优化 | Solve(options, &problem, &summary); |

Tips: operator () 函数要有一个const 后缀， operator 内的常数数字要在T()内

Options参数 <https://blog.csdn.net/DumpDoctorWang/article/details/84890792>

1. C++ 语法细节

|  |  |
| --- | --- |
| 1. 智能指针shared\_ptr | 普通指针的转化只能通过显式的方式  make\_shared<int> 是一个模板 后面要有一个<>  .release(); |
| 1. CMakeLists.txt | Add\_executable () 要把文件所有.cpp 文件加进来 不然.cpp 的内容不会进行编译 |
| 1. Set unordered\_set 和Eigen冲突 | 学Cartographer 使用bitset进行set 的存储 |
| 1. Mapbuilder |  |
| 1. Vector 拼接 | vector a; vector.b; a.insert(a.end(),b.beign(),b.end()); |
| 1. C++ 类型转换 | <https://www.cnblogs.com/chenyangchun/p/6795923.html> |
| 1. C++ 计时chrono | <https://blog.csdn.net/DumpDoctorWang/article/details/81781227>  <https://blog.csdn.net/bao_bei/article/details/77162321> |
| 1. C++ 多态 |  |
| 1. 调试上VS 好用，Linux 快   <https://stackoverrun.com/cn/q/10202721> VS /02和/RCT1 出现冲突 | |
| 1. Can not declare the member function to have static linkage 也就是说 static member function 在.cc 中进行定义时不需要再添加static 关键字 | |
| 1. 过于 LNK 2001 静态成员上的错误： 静态成员需要进行类外定义， 类内的那个只是一个声明 | |
| 1. 之后 undefined reference to WinMain 16 是因为 静态成员的类外定义没有和静态成员的定义放在一个文件内   之前一个放在.h 一个放在.cpp里了 | |

1. 实现细节

V01

|  |  |
| --- | --- |
| 1. VoxelFilter 结合 unordered\_set 和bitset来完成 | * Eigen：：Array2d和set 和unordered\_set 犯冲 移位操作<<完成 |
| 1. OpenCV点云显示 | * DrawPoint 函数来写 |
| 1. GetAllData（） | * 信息txt文件 应该放在可执行文件的目录下 |
| 1. ComputeOccGrid->distanceTransform | distanceTransfrom+ threshold实现比较繁琐  所以直接采用自己写的self-distanceTransform  Mat::copytTo 有很多坑 第二个参数一定要是一个CV\_8UC1不然内存泄漏 最后用CV\_16FC1代替，并且写了一个和函数将self\_distanceTransform实现Mat 访问元素时的指针需要多注意，如果类型步匹配也会出问题  性能差了3700倍还是使用opencv 的函数了(实现的细节上水平差太多了)  Matlab round +1 因为matlab 矩阵从1开始计数  C++ 从0 开始计数 |
| 1. Kdtree | 报错在.obj 中有定义LNK2005 因为没有把声明和定义放在了一起导致的  更改后效果图：回环误差加大，猜测是之后添加voxel\_filter 让地图计算不是那么准导致的，毕竟这里用的voxel\_filter 没有取质心，对于地图的形状还是有所改变。会在向kdtree插入node的时候偶尔出现卡顿，kdtree 更适合在已有地图上进行定位使用，而不是在构建地图的时候存储地图点。但是在搜索时的确是可以加速点的搜索。 |
| 1. Drawer OpenCV 2d 显示 | DrawerPoint TESTPASS |

效果分析

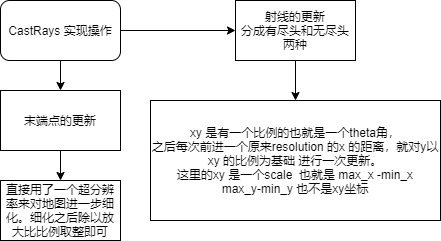
|  |  |
| --- | --- |
| 效果 3000帧 | 效果5500帧 |
|  |  |
| 修改代码后的效果 |  |
|  | 原本错误的地方：   1. 做DownHill 搜索的时候，旋转搜索的时间，直接在LocalMap系中进行了，而没有在世界坐标系下进行。 第一版 2. 一种是先做一个delta\_的微小运动，然后在做guess\_pose.   一种是先做一个guess\_pose 然后做一个微小运动。  区别是一个delta 的运动矩阵左乘一个是右乘。  但是搜索类算法的想法是直接找一个最合适的guess\_pose和微小运动结合成的运动矩阵，也就是最合适的xy theta 三个参数。  错误原因： 最开始的一半想要寻找一个微小运动然后和guess\_pose 结合算出结果(也就是先把点从guess\_pose 坐标投射到微小运动的坐标内，然后寻找最优微小运动参数)，但是因为这种矩阵乘法表现在运动参数(xytheta)上时，是一个非线性的相加。所以最后在返回值的时候我让 两个运动的参数直接相加的方式，必然会导致平移方面的误差(xy; 因为是2d 的原因)。 也就是说最后最优的参数 是从两个运动矩阵相乘后，得到的参数值，而不是一个粗暴地将两个运动矩阵的参数相加。  正确的方法：直接在guess\_pose 的三个参数上进行搜索，这会让运动在李代数上做每次不等值的变换，但是在R3 (x y theta)上做等值的变换。省掉了guess\_pose 往微小运动坐标系的变换，找到的直接就是最优参数。  这里可以说也是一个对于Cartographer的小改进 Cartographer 用的是后者, 我用的是前者 |
| 结果分析&后期改进   1. 在Matlab 里面实现的时候，可以发现HillClimb 是一种对变化非常敏感的搜索方式。但是速度快！！HillClimb可以作为一种subpixel 的方式来提高搜索的精度，猜测是可以代替Ceres 来进行搜索的。而且HillClimb在对odd 地图的适应性上比ICP 要强。 2. 从结果可以看出来这个里程计的精度已经算是不错的，因为每一次点的累计只是累计了新的点。所以误差增加很慢。猜想一旦地图中出现一个绝对路标进行矫正，误差不会太大。 3. 为了保证项目是一个SLAM系统，而不是一个里程计，在此基础上添加后端优化的过程 4. HillClimb 最致命的问题是如果这个地图中存在移动的物体，那么因为算法对变化的敏感性，那么很容易出问题。所以还是决定先用Odd 来代替这个地图，然后想办法把HillCLimb和odd地图结合起来，降低对动态地图的敏感性。 5. 在进行回环检测的时候Cartographer 的分支定界明显是比HillClimb暴力匹配更好的方法。一个是全局最优一个是局部最优，而且在速度上会有很大的差别。但是Cartographer 的回环检测也是可以再优化一点点的，关键在于Cartographer 在分支定界的时候要保存一个现有最大结果，这在整数规划的时候是有体现的。但是在Cartographer v1.0 版本中并没有体现。 | |

V02

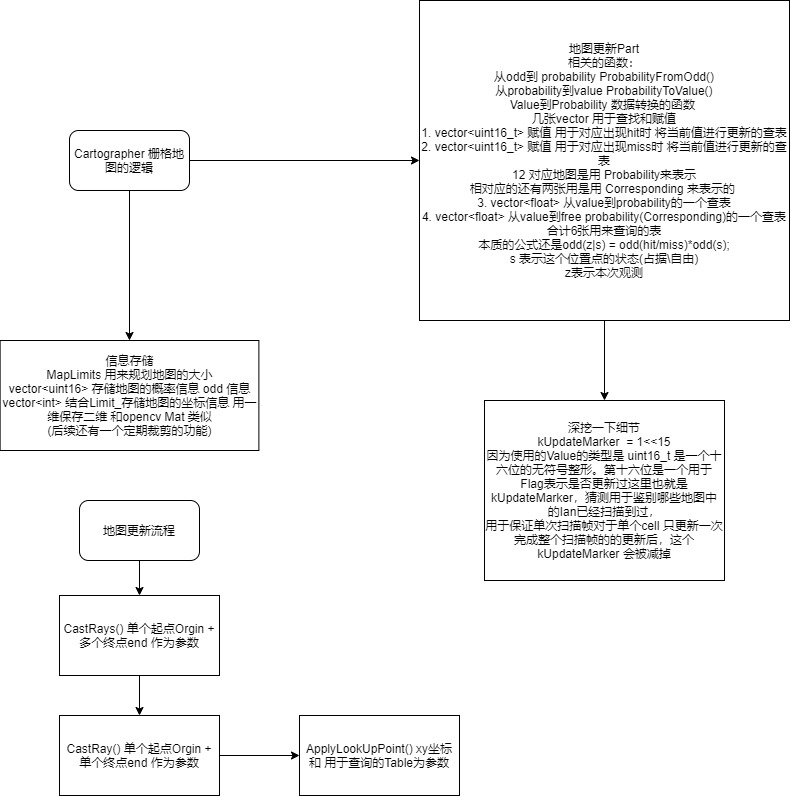
|  |  |
| --- | --- |
| 1. 重构代码 | 各个类支持各个功能 |
| 1. 添加占据栅格地图 | * PGMBuilder ☑在构造函数时，进行图的初始化。 |
| * GrowAsNeeded()☑使用 Eigen::AlignedBox 来进行拓展   AlignedBox3d 的例子 .extend 用于扩展  .min .max 提取最大最小值 |
| * Submap() 拼接 ☑: 每一个子图有一个origin 当做子图的坐标，用于之后检索每一帧和子图的位姿。然后每一个子图都有一个AlignedBox2i 来确定这个子图在整体子图的位置 AlignedBox .min和.max 都是直接可以以离散的世界坐标进行保存   RayCast 划线 ☑  ApplyLookUpPoint()☑  一个概率查询表 ☑  表的预计算 ☑  Odd 和 概率数值转化 ☑  Debug:   1. 在测试Insert的时候出现问题，问题就是估计出来的数值和实际使用的值是一个相反的关系，经过检查之后，发现原因如下：在处理新一帧时，在匹配的时候，做的操作可以理解把新的一帧的坐标进行旋转平移，和原本的地图进行匹配。所以这样得出的微小运动的估计值和实际的微小运动运动是一个相反的值。   测试通过☑   1. 集成到main.cpp中进行估计 2. pixelSize =0.1 效果没有问题 pixelSize = 0.05 结果螺旋爆炸。猜测:离散化的时候有的参数是直接赋的数值。 原因：SearchWindow的 pixelSize参数在GenerateRotDiscre 的时候用来离散化了，就影响了离散点的坐标。 3. 第二个问题是会在62 到64 帧的时候出问题，运动会超过预设值。因为，一直没有往内部进行插值，也就是说MotionFilter设置的太大了。原因：出现了1/0的情况 4. 虚拟机出问题无法进行Ceres测试进行精匹配   用了大量的时间，在debug这里，现在是明白里面的坑了  **后期改进：需要把 CSM::Match 函数里面的TODO 完成！** |
| 1. 多分辨率搜索地图 | * 高分辨率往低分辨率的转换☑   二维地图转一维的缘故，先处理x方向，然后多个x方向一起处理y方向，为了提高复用率往向前移动一个格子的时候里面有很多值都没有改变，这就是Cartograpeher的一个实现的细节。  实现的方式和我用Matlab实现的方式不一样实现的非常好。  具体实现使用SlideWindow 运动deque 来实现。  具体细节：因为之后做分支定界的时候，大小确定的方式是[To,To + delta\_T] 所以在做概率占据地图的时候需要考虑，整个模糊的方向的问题。  小项目中采取 从左往右 从上往下取最大的方向。   * 确定低分辨率时的搜索范围 ☑ * 还需要优化一下地图存储的模式 * Debug：10点了 发现问题是因为在构建地图的时候没有对地图进行扩大导致信息没有补充完全导致的 |
| 1. 分支定界C++转换 | 和CSM的搜索不同。 CSM是一个在已有平移坐标系的情况下进行微调，但是在BranchAndBound的中为了更大地进行扫描，就要换成后者，再来进行计算。还是有一些细节上要看一下  修改的内容   * M2MPGM：各种get函数 ☑ * CSM 添加BBM (分支定界的部分) 还差递归的部分☑ * 打分函数对于外点 如何确保函数运作正常 ☑   这个问题在前端submap 的时候是不会出现的，到了后端因为大位移的出现所以就需要调整一下   * CSM 的BBM 里面的ShrinktoFit 看一下 Cartographer 的转换。☑ **进一步缩小搜索范围**   ShrinktoFit 每一个Node 对于map 大小来确定能左右上下移动多少，来生成相对少量的Candidates。通过看代码ShrinkToFit在大地图成功率更高。也就是说因为ShrinkToFit 无法每次都很精准的进行匹配，所以回环是有成功率的，而不是百分百成功。   * 还没有添加一个硬性的threshold 已经添加好☑ 但是还没有具体设置值 * BBM submap 并没有使用拼接，来做总体的匹配，而是用每一个Node 和 每一个submap 来进行匹配。☑ * Debug  1. 优化可能的candidates 从5s 到1s 。**TODO: 还是有大量无用的值，下次优化的时候，计算点云的均值和方差，然后根据这个特点来进一步消除冗余candidates。** 2. 主要问题是在多层地图生成上，要在地图的左边和上边两边各生成stride -1个格子，第一次单scan 已经通过。但是角度上差了4度。弧度差了0.07 还是有问题。但是基本8m内的回环都可以检测到。但是20m的单帧回环就没有办法了。按理来说一个通了应该所有都可以，这里还需要再注意一下。全部可以跑通，之前的问题是因为BBM\_Loop 中每次切分成四个candidates 的时候，划分stride 的符号写错了， 应该和制作多层地图时取最大值的方向相对应。 3. BBM加速改成/o2之后速度加速300倍。基本可用 4. 已经在surface上逻辑推导画图证明基本可以用。 |
| 1. Ceres优化 | Bicubic interpolation  <https://blog.csdn.net/qq_24451605/article/details/49474113>  双三次差值 差值的范围是从0到1 但是如果不知道导数值的话，就行只能从[0,1]范围左右分别多取一个点来进行一阶导数的近似。 |
| 1. PoseExtraPolator | 互补滤波：PoseExtrapolator  使用的方法来自<https://www.zhihu.com/people/hu-xiao-dao-51/answerss> 无人机专栏--无人机姿态解算---适应互补滤波  Roll Pitch Yew 先得到前两个值，所以yew 会无法得到 |
| 1. 线程池 | <https://wangpengcheng.github.io/2019/05/17/cplusplus_theadpool/>  <https://www.cnblogs.com/ailumiyana/p/10016965.html>  <https://www.cnblogs.com/yangang92/p/5485868.html> |
| 1. OpenCV 画图适应odd |  |
| 1. 后期优化 | TODO LIST:   1. **还是有大量无用的值，下次优化的时候，计算点云的均值和方差，然后根据这个特点来进一步消除冗余candidates。** 2. **熟悉一下shared\_ptr 保证没有内存泄漏；** 3. **所有的函数只要是传大量的值返回必须是传引用或者指针**   **---------------------------前三项是最急切的要求---------------------------------------**   1. **找到大耗时的function 优化函数实现 现在已经知道的有CastRays和BBM** 2. **后期优化 Cartographer 的位姿融合改成ESKF imu和 odometry 的紧耦合** 3. **发挥CSM的方差让前端变成一个imu和odomettry和CSM 的EKF耦合估计** 4. **为了避免多次传值的voxel\_filter\_scan 使用new 和delete来管理点云内存，防止在AddKeyScan的时候传值** 5. **第7的情况在很多函数中，传值的情况还是存在，需要后期全部换成，传引用或者传指针** |
| 1. Vector resize 和reserve 的 排查 | PMG CSM 没有添加BBM 的版本已经TEST PASS |
| 1. MapLimits 的Contain函数的scaleXY 部分要检查一下 | +1 -1 的问题 也是 0 和1 谁是最小的问题 直接包含在8 项内一起检查过  并且TEST PASS |
| 1. LaserFan 的数据是全局坐标还是局部坐标要查一下 | 内部的pose是全局坐标，laserdata 是pose的局部坐标系内 |

Odd 地图逻辑

根据射线的odd地图更新

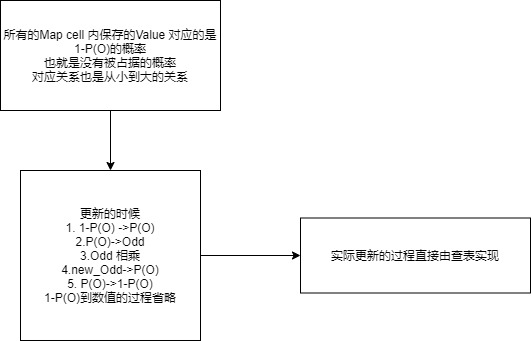
 **已经依据这个逻辑写了一个我自己的逻辑**

Cartographer 栅格地图更新逻辑



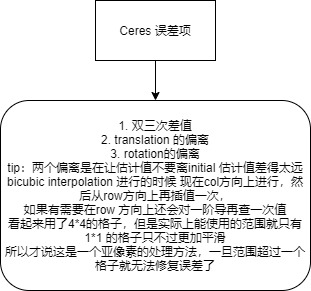
odd占据数值逻辑

<https://zhuanlan.zhihu.com/p/49030629>

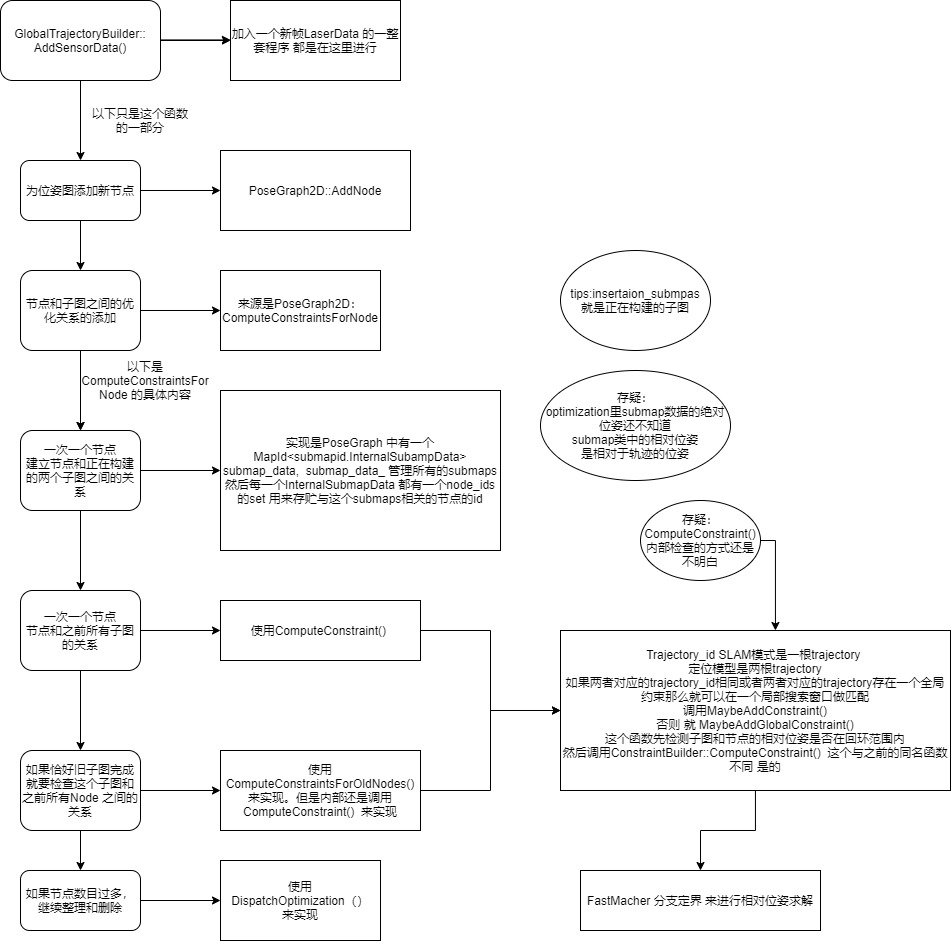


数值优化与位姿图

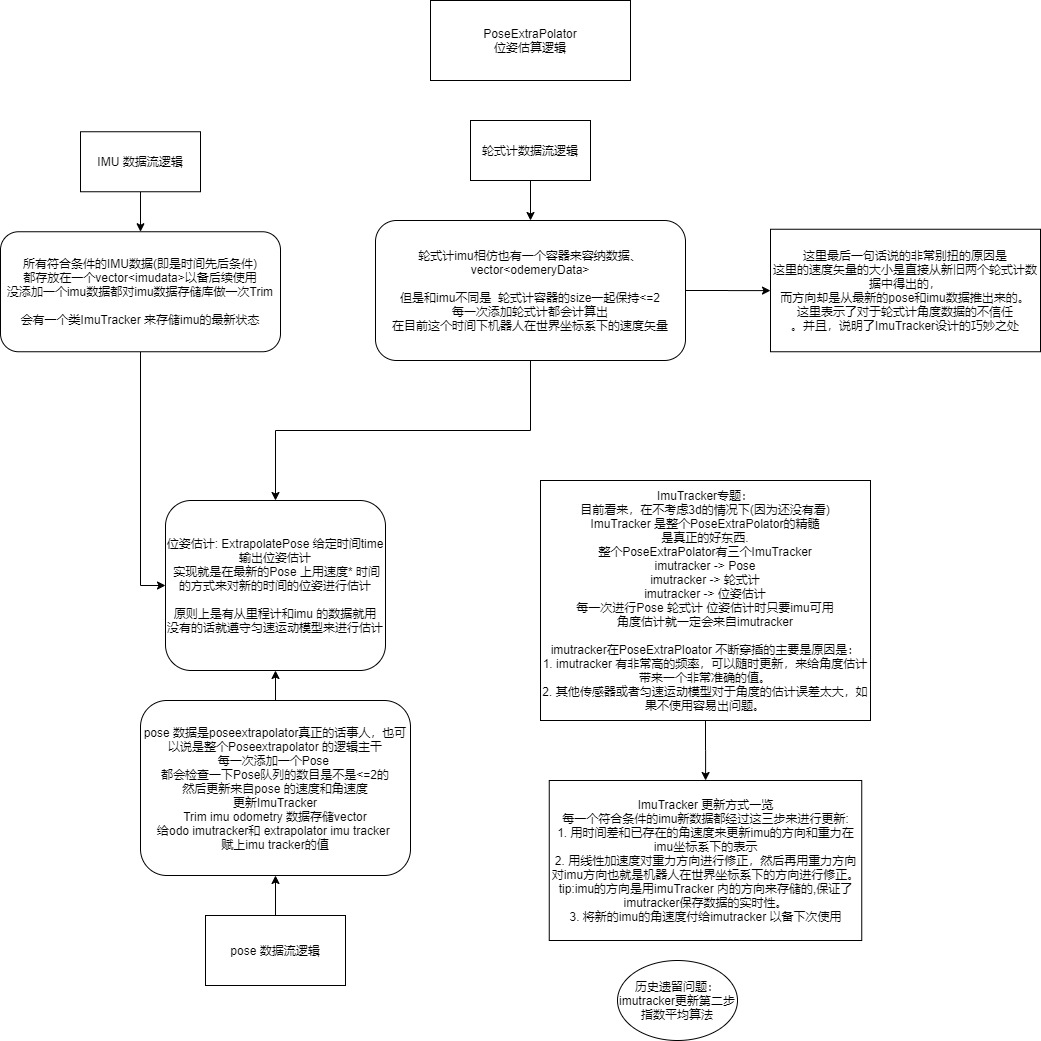
Ceres 的bicubic interpolation



位姿图约束添加逻辑



Cartographer 融合逻辑PoseExtraPolator



🡪后期优化

OpenCV编程入门 C++

|  |  |
| --- | --- |
| 1. Opencv 安装细枝末节 Windows | <https://blog.csdn.net/weixin_43503473/article/details/103644313> + OpenCV 编程入门P27  注意电脑和编译器是x86 还是x64 |
| 1. OpenCV imread 各种类型的图像 | <https://www.learnopencv.com/read-an-image-in-opencv-python-cpp/> |
| 1. OpenCV Mat元素访问 | <https://www.cnblogs.com/kuotian/p/6389260.html> |
| 1. OpenCV Mat 成员变量解读 | <https://www.cnblogs.com/wangguchangqing/p/4016179.html> |
| 1. OpenCV ROI感兴趣区域操作 | 1. OpenCV3 编程入门(无目录) P132-133 有目录115-116 两种方法来读取一个矩阵块   CopyTo() 来实现   1. AddWeight() 线性混合的部分没有再看了   <https://www.cnblogs.com/skyfsm/p/6892746.html>  **Rect()有大坑 它的参数 排列不是row cols 而是 xy 也就是 cols row 的排列**  **2020.3.7 补充：记得是和mat实例化的顺序不同所以debug了很久才发现** |
| 1. OpenCV distanceTransform | 0 代表物体 非零值代表背景 |
| 1. OpenCV clone、copyTo | <https://blog.csdn.net/DreamLike_zzg/article/details/79169336>  <https://blog.csdn.net/u013270326/article/details/72730812>  clone 和 copyTo的区别 copyTo 第二个参数mask 中的非0会保留 0 值会变透明忽略  copyTo 的第二个参数只能是CV\_8UC1 不然内存泄漏 要自己写一个函数来进行矩阵和运算 |
| 1. OpenCV 画图 | <https://blog.csdn.net/chaipp0607/article/details/56277479>  几个画图的函数  <https://tool.oschina.net/commons?type=3>  RGB 色号 |
| 1. OPencv Mat to vector 转换 | <https://blog.csdn.net/qq_27278957/article/details/88652661> |