

子地图管理类结构设计

|  |  |
| --- | --- |
| 文档版本 | V 1.0.0 |
| 发布日期 | 2022-07-04 |

**版权所有©舵敏智能2021，保留一切权力。**

非经本公司书面许可，任何单位或个人不得擅自摘抄、复制部分或全部本文档，不得以任何形式传播。

**商标声明**

、舵敏、舵敏智能和其他舵敏商标均为舵敏智能科技的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人所有。

**上海舵敏智能科技有限公司**

**舵敏智能科技（苏州）有限公司**

|  |  |
| --- | --- |
| 网址: | https://www.dominant-tech.com/ |
| 客户服务邮箱: | support@dominant-tech.com |

**目 录**

[1、Atlas管理地图块 1](#_Toc1381729652)

[1.1 Atlas简介 1](#_Toc700894516)

[1.2 与ORBSLAM3中Atlas类的区别 1](#_Toc2028196659)

[2、地图点块、关键帧块信息结构体设计 2](#_Toc1669879627)

[2.1关键帧块信息表与地图点块信息表 2](#_Toc1857635829)

[3、子地图管理类设计 3](#_Toc162479934)

[3.1 需要维护的变量 3](#_Toc987927284)

[3.2 需要的功能函数 4](#_Toc505161466)

[4、子地图类设计 5](#_Toc2033546)

[4.1 需要维护的变量 5](#_Toc981830024)

[4.2 需要的功能函数 6](#_Toc1634634359)

## **1、Atlas管理地图块**

## 1.1 Atlas简介

Atlas为地图集，用于管理关键帧块和地图点块。实现对关键帧块和地图点块的一些操作，如下：

|  |  |
| --- | --- |
| **创建**关键帧块和地图点块 | **融合**关键帧块和地图点块 |
| **存储**关键帧块和地图点块 | **加载**关键帧块和地图点块 |

ORBSLAM2只有一个地图Map，而ORBSLAM3的Map作为子地图被Atlas管理。相比于ORBSLAM2的Map类，新增了一些成员。进而体现与ORBSLAM2的不同，ORBSLAM2中，丢失后需要回到原位置进行重定位。但ORBSLAM3中，如果tracking线程丢失，会在之前的所有子地图中进行查询和匹配，如果匹配成功，则tracking线程继续。匹配不成功，则重新创建新的子地图继续跟踪。此外，构建多个子地图可以降低误差的累计。

## 1.2 与ORBSLAM3中Atlas类的区别

实现地图分块的目的：**在数据层面实现分块，减少内存占用、节省资源；**

|  |  |  |  |
| --- | --- | --- | --- |
| **Atlas** | | | |
|  | **ORBSLAM3** | **Our** | **注** |
| **管理对象**  **（操作对象）** | 当前地图 | 当前地图  地图点块、关键帧块 | 我们的Atlas操作的对象对更底层 |
| 子地图 |
| **对象创建条件** | 跟踪丢失时保存子地图，新建子地图继续跟踪 | 根据关键帧坐标、地图点坐标创建关键帧块、地图点块 | OBSLAM3在功能上实现子地图在数据上并不算分块（子地图大小不确定） |
| **保存时机** | 系统结束前统一保存 | 增量式保存 | ORBSLAM3的子地图一直在内存中 |
| **加载时机** | 系统开始前全部加载 | 根据当前位置等因素动态加载 |  |

我们管理对象的更底层，分块更底曾。对于ORBSLAM3的子地图来说，可以在子地图的基础上结合关键帧块，地图点块，实现子地图形式的分块存储。

## **2、地图点块、关键帧块信息结构体设计**

## 2.1关键帧块信息表与地图点块信息表

|  |  |
| --- | --- |
| **块坐标结构体** | |
| 地图点块坐标（MPBlockPos） | 关键帧块坐标（KFBlockPos） |
| int x; | int x; |
| int y; | int y; |

|  |  |  |
| --- | --- | --- |
| **块信息表结构体** | | |
| **地图点块信息表**  **（MPBlockTable）** | **关键帧块信息表**  **（KFBlockTable）** | **类型** |
| 地图点块ID | 关键帧块ID | string（MPB\_4\_31、KFB\_1\_12） |
| 地图点块中心坐标 | 关键帧块中心坐标 | MPBlockPos、KFBlockPos |
| 地图点总数 | 关键帧总数 | int |
| — | vector<地图块信息表> | vector |
| 创建时间 | 创建时间 | double |
| 更新时间 | 更新时间 | double |

**2.2关键帧块类、地图点块类设计**

|  |  |  |
| --- | --- | --- |
| **关键帧块类** | | |
| **成员变量类型** | **成员变量名** | **说明** |
| KFBlockPos | mKFBlockPos | 关键帧块的中心坐标 |
| KFBlockTable | mKFBlockTable | 关键帧块信息表 |
| short int | mKFBlockScale | 关键帧块的尺寸 |
| set<KeyFrame\*> | mspKeyFrames | 关键帧容器 |
| BlockDatabase\* | mBlockDB | 关键帧块所属数据库 |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **成员函数** | | | **功能** | **参数说明** |
| **函数名** | **输入** | **输出** |
| AddKeyFrame | KeyFrame\* | void | 向关键帧块中增加关键帧 | 输入：关键帧指针 |
| ComputeMPBlock | void | void | 计算关键帧块观测到的地图点块 | — |
| SetKFBlockTable | string ID,  double CreateTime, double UpdateTime | void | 查找关键帧块在临时关键帧块列表中是否存在 | 输入：关键帧块ID  块创建时间  块修改时间 |
| GetKFBlockTable | void | KFBlockTable\* | 获得关键帧块的信息表 | 输出;关键帧信息表指针 |
| DeleteKeyFrames | void | void | 删除关键帧块的关键帧 | — |
| GetKeyFrameNums() | - | unsigned long | 在系统结束时获取保存的关键帧数量 |  |

|  |  |  |
| --- | --- | --- |
| **地图点块类** | | |
| **成员变量类型** | **成员变量名** | **说明** |
| MPBlockPos | mMPBlockPos | 地图点块的中心坐标 |
| MPBlockTable | mMPBlockTable | 地图点块信息表 |
| short int | mMPBlockScale | 地图点块的尺寸 |
| set<MapPoint\*> | mspMapPoints | 地图点容器 |
| BlockDatabase\* | mBlockDB | 地图点块所属数据库 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **成员函数** | | | **功能** | **参数说明** |
| **函数名** | **输入** | **输出** |
| AddMapPoint | MapPoint\* | void | 向地图点块中增加地图点 | 输入：地图点指针 |
| SetMPBlockTable | string ID,  double CreateTime, double UpdateTime | void | 查找地图点块在临时地图点块列表中是否存在 | 输入：地图点块ID  块创建时间  块修改时间 |
| GetMPBlockTable | void | MPBlockTable\* | 获得地图点块的信息表 | 输出：地图点信息表指针 |
| DeleteMapPoints | void | void | 删除地图点块中的地图点 | — |
| GetMapPointsNums | - | unsigned long | 在系统结束时获取保存的地图点数量 |  |

## **3、子地图管理类设计（Atlas）**

## 3.1 需要维护的变量

|  |  |  |
| --- | --- | --- |
| **成员变量类型** | **成员变量名** | **说明** |
| short int | mKFBlockScale | 关键帧块尺寸 |
| vector<KFBlockPos> | mKeyFrameBlockTempID | 存放临时关键帧块坐标，方便查找 |
| vector<KeyFrameBlock\*> | mKeyFrameBlockTemp | 存放临时关键帧块 |
| vector<KFBlockPos> | mKeyFrameBlockWaitingID | 存放待存储关键帧块坐标，方便查找 |
| vector<KeyFrameBlock\*> | mKeyFrameBlockWaiting | 存放待存储关键帧块 |
| KeyFrameBlock\* | mKeyFrameBlockLoading | 存放加载的关键帧块 |
|  |  |  |
| short int | mMPBlockScale | 地图点块尺寸 |
| vector<MPBlockPos> | mMapPointBlockTempID | 存放临时地图点块坐标，方便查找 |
| vector<MapPointBlock\*> | mMapPointBlockTemp | 存放临时地图点块 |
| vector<MPBlockPos> | mMapPointBlockWaitingID | 存放待存储地图点块坐标，方便查找 |
| vector<MapPointBlock\*> | mMapPointBlockWaiting | 存放待存储地图点块 |
| MapPointBlock\* | mMapPointBlockLoading | 存放加载的地图点块 |
| BlockDatabase\* | mBlockDatabase | 所属的数据库指针 |
| Map\* | mpMap | 所属地图 |
| unsigned long int | mStratKeyFrameID | 检测窗口起始位置 |
| int | mWindowSize | 实际窗口移动大小 |
| int | RangeFactor | 范围参数 |
| float | factor | 关键帧块扩展边长, 防止在块边界蛇行运动,导致频繁删除 |
| vector<MapPoint\*> | vpDeleteMPs | 存放要删除的地图点 |
| vector<KeyFrame\*> | vpDeleteKFs | 存放要删除的关键帧 |
| bool | mbReuse | 是否复用地图 |
| Tracking\* | mpTrack | Tracking指针 |
| LocalMapping\* | mpLocalMapping | LocalMapping指针 |
| KeyFrameDatabase\* | mpKeyFrameDatabase | KeyFrameDatabase指针 |
| ORBVocabulary\* | mpVocabulary | ORBVocabulary指针 |
| std::list<KeyFrame\*> | mlKeyFramesWaitCheck | Atlas待检测关键帧列表 |
| string | mStrSaveToFile | 存储、加载块的路径 |
| bool | mbFinishRequested | 是否有停止请求 |
| bool | mbFinished | 线程是否完成 |
| std::mutex | mMutexAtlas | 互斥锁 |
| std::mutex | mMutexFinish | 互斥锁 |
| Modified by zwx 2023.06.05 | | |
| vector<KeyFrame \*> | mvpKFErase | 在删除关键帧和地图点时，记录可以删除的关键帧和地图点。之后进行统一删除，避免多次访问Map导致频繁加锁。 |
| vector<MapPoint \*> | mvpMPErase |
| 保留小地图点块，减少IO次数，避免频繁的块融合 | | |
| map<MPBlockPos, MapPointBlock \*> | mMPBlocksReserve | 记录地图点数量没有达到阈值的小地图点块 |
| map<string,MapPointBlock \*> | mSmallMPBsLoads | 保存加载上来的小地图点块 |
| 缓存加锁变量，避免频繁加锁 | | |
| unsigned long | mLocalMinMPId, mLocalMinKFId, mMapMaxKFid | 记录局部地图中的最小地图点Id、最小关键帧Id、最大关键帧Id |
| 保存关键帧数据库 | | |
| std::map<unsigned long, cv::Mat> | mDesOfSavedKF | 关键帧对应的特征描述符 |
| std::map<unsigned long, vector<cv::KeyPoint>> | mKeysUnOfSavedKF | 关键帧的特征点 |
| std::map<unsigned long, KFBlockPos> | mBlockIdOfSavedKF | 关键帧存储的块Id |
| std::map<unsigned long, DBoW2::BowVector> | mBowOfSavedKF | 关键帧的词袋向量 |
| std::map<unsigned long, DBoW2::FeatureVector> | mFeatOfSavedKF | 关键帧的特征向量 |
| std::vector<list<unsigned long>> | mvInvertedFile | 恢复的关键帧数据库结构 |
| 其它 | | |
| const static int | mSafetyInterval | Atlas保存窗口和局部地图之间预留的安全区间（KF个数），避免离开了局部地图的关键帧又重新进入局部地图，导致删除关键帧后程序 bug |
| vector<MapPointBlock \*> | mMapPointBlockMerging | 记录加载地图点时创建的地图点块 |
| vector<KeyFrameBlock \*> | mKeyFrameBlockMerging | 记录加载关键帧时创建的关键帧块 |
| vector<MapPoint \*> | vpTryDeleteMPs | 记录已保存待删除的地图点 |
| vector<KeyFrame \*> | vpTryDeleteKFs | 记录已保存待删除的关键帧 |
| std::condition\_variable | mConditionVarWaitingKF | Atlas等待LocalMapping插入关键帧 |
|  |  |  |

## 3.2 需要的功能函数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **成员函数** | | | **功能** | **参数说明** |
| **函数名** | **输入** | **输出** |
| **KeyFrame** | | | | |
| CreateKeyFremeBlock | KFBlockPos，int | void | 创建临时关键帧块 | KFBlockPos:块坐标  int：块尺寸 |
| AddKeyFrameToBlock | vector<KeyFrame\*> | vector<KFBlockPos> | 遍历关键帧将其添加到所属关键帧块 | 输入：关键帧数组  输出：关键帧块数组 |
| IsKeyFrameBlockTemp | const KFBlockPos | int | 查找关键帧块在临时关键帧块列表中是否存在 | 输入：关键帧块坐标  输出：关键帧块位置 |
| MergeKeyFrameBlock | void | void | 将临时块和已经存储的块融合 | — |
| KeyFrameBlockSave | void | void | 保存关键帧块 | — |
| KeyFrameBlockLoad | string | bool | 加载关键帧块 | 输出：是否加载成功 |
| CheckKeyFrameSave | KeyFrame\*, bool | void | 检测关键帧是否可以保存 | 输入：当前使用的参考关键帧，系统停止标志 |
| DeleteKeyFrames | vector<KeyFrame\*> | void | 删除已经存储的关键帧序列 | 输入：要从内存中删除的关键帧序列 |
| **MapPoint** | | | | |
| CreateMapPointBlock | MPBlockPos，int | void | 创建临时地图点块 | MPBlockPos:块坐标  int：块尺寸 |
| AddMapPointToBlock | vector<MapPoint\*> | vector<MPBlockPos> | 遍历地图点将其添加到所属地图点块 | 输入：地图点数组  输出：地图点块数组 |
| IsMapPointBlockTemp | const MPBlockPos | int | 查找地图点块在临时地图点块列表中是否存在 | 输入：地图点块坐标  输出：地图点块位置 |
| MergeMapPointBlock | void | void | 将临时块和已经存储的块融合 | — |
| MapPointBlockSave | void | void | 保存地图点块 | — |
| MapPointBlockLoad | string | bool | 加载地图点块 | 输出：是否加载成功 |
| CheckMapPointSave | vector<KeyFrame\*> | void | 根据关键帧获得要要分块的地图点 | 输入：关键帧数组 |
| DeleteMapPoints | vector<MapPoint\*>, bool | void | 删除已经存储的地图点序列 | 输入：要从内存中删除的地图点序列,系统停止标志 |
|  |  |  |  |  |
| CheckBlocksLoad | vector<float>, int | vector<string> | 查询前左右方向相邻的关键帧块 并筛选 | 输入：  当前位置, 范围系数 |
| BlocksLoad | vector<string> | void | 加载前左右方向相邻的关键帧块 以及 关键帧块观测到的地图点块 | 输入：要加载的关键帧块 |
| GetKeyFrameBlocksAround() | vector<float>, int , float | void | 根据当前位置获取周围所有的关键帧块ID | 输入：当前位置,范围系数,当前所在关键帧块的扩大范围 |
| DeleteKeyFrameBlocks | vector<float>, int , float | void | 删除不再需要的关键帧块 | 输入：当前位置,范围系数 |
| DeleteMapPointBlocks | vector<float>, int , float | void | 删除不再需要的地图点块 | 输入：当前位置,范围系数 |
| isCurrKFBlockPro | vector<float>, float | bool | 判断位置是否在当前关键帧块的扩大范围内 | 输入：当前位置,范围系数 |
| AddKeyFrameToCheck | KeyFrame\* | void | 向Altas线程添加关键帧 | 输入：关键帧 |
| ReconstructMap | vector<KeyFrame\*>, vector<MapPoint\*> | void | 恢复关键帧、地图点之间的关系 |  |
| CheckWaitKeyFrames() | void | bool | 检查Atlas是否还有待检测关键帧 |  |
| SaveBlockDatabase()  LoadBlockDatabase() | void | void | 保存、加载块数据库 |  |
| Run() | void | void | 主函数 |  |
| CheckFinish() | void | bool | 同ORBSLAM2 |  |
| isFinished() | void | bool |  |
| SetLocalMapper() | LocalMapping | void |  |
| SetTracker() | Tracking | void |  |
| ResetIfRequested() | void | void |  |
| RequestReset() | void | void |  |
| RequestFinish() | void | void |  |
| SetFinish() | void | void |  |
| 保留小地图点块 | | | | |
| ReserveSmallBlock() | - | void | 保留地图点数量较少的小地图点块 |  |
| AddSmallBlock() | MPBlockPos | void | 从小地图点块中选择地图点块进行添加代替创建新的地图点块 |  |
| SaveSmallBlock() | - | void | 在系统结束时，将所有未保存的小地图点块合并为一个大块并保存 |  |
| LoadSmallBlock() | - | void | 加载小地图点块 |  |
| 保存关键帧数据库 | | | | |
| AddBowDatabase() | vector<KeyFrame \*> & | void | 添加要保存的关键帧数据 | 输入：关键帧数组 |
| SaveBowDatabase() | - | void | 保存关键帧数据库 |  |
| LoadBowDatabase() | - | void | 加载关键帧数据库 |  |
| RestoreBowDatabase() | - | void | 恢复关键帧数据库结构 |  |
| 删除磁盘关键帧 | | | | |
| DeleteMapBlock() | int num | void | 当磁盘空间不够时，删除保存的前n个关键帧及其对应的地图点 |  |
| 其它 | | | | |
| extractWaitKeyFrames() | - | std::list<KeyFrame \*> | 拷贝待检查的关键帧变量到本地 | 输出：拷贝的关键帧数组 |
| DeleteLoadingBlocks() | - | void | 删除加载地图点时创建的关键帧块和地图点块 |  |
| AtlasWaitingKF(); | - | void | Atlas等待LocalMapping插入关键帧 |  |

## **4、地图类设计**

## 4.1 需要维护的变量

|  |  |  |
| --- | --- | --- |
| **类型** | **功能** | **说明** |
| vector<KeyFrame\*> | 保存了最初始的关键帧 | 优化时作为固定帧 |
| vector<unsigned long int> | 该子地图后续关键帧的ID |  |
| std::mutex | 地图更新互斥锁 | 地图更新时加锁 |
| std::mutex | 地图点创建互斥锁 | 避免创建两个点相同的点 |
| std::mutex | 对地图的操作需要互斥 |  |
| static long unsigned int | 下个地图编号 | 在当前地图编号上自增1，新的地图构造后，直接将其赋新地图 |
| long unsigned int | 地图编号 |  |
| set<MapPoint\*> | 地图的地图点集合 |  |
| set<KeyFrame\*> | 地图的关键帧集合 |  |
| KeyFrame\* | 地图的第一个关键帧 | 用于优化 |
| KeyFrame\* | 地图的最小ID的关键帧 | 用于检索所有子地图的最小关键帧ID，统计丢失帧数目 |
| unsigned long int | 地图第一、最小、最大关键帧ID |  |
| bool | 地图状态标记 | 状态有活跃、不活跃两种状态 |
| bool | 地图是否是bad | 目前知道融合后需要将当前地图设为bad |
| unsigned long int DeleteKFID | 关键帧删除分界 |  |
| unsigned long int DeleteMPID | 地图点删除分界 |  |

## 4.2 需要的功能函数

|  |  |  |  |
| --- | --- | --- | --- |
| **输入** | **输出** | **功能** | **说明** |
| 关键帧ID | void | 构造新的地图 | 参数为该地图初始化的第一个关键帧 |
| void | void | 无参构造 | 构造第一个地图 |
| void | void | 析构函数 |  |
| KeyFrame\* | void | 增加关键帧 |  |
| MapPoint\* | void | 增加地图点 |  |
| KeyFrame\* | void | 删除关键帧 |  |
| MapPoint\* | void | 删除地图点 |  |
| void | vector<KeyFrame\*> | 获取所有关键帧 |  |
| void | vector<MapPoint\*> | 获取所有的地图点 |  |
| void | long unsigned int | 获取关键帧的数目 |  |
| void | long unsigned int | 获取地图点的数目 |  |
| void | long unsigned int | 获取地图点ID |  |
| void | long unsigned int | 获取地图最大关键帧ID |  |
| void | unsigned int | 获取地图最小关键帧ID |  |
| void | long unsigned int | 获取地图初始关键帧ID |  |
| long unsigned int | void | 设置地图初始关键帧ID |  |
| void | KeyFrame\* | 获取初始关键帧 |  |
| void | void | 设置当前地图为活跃状态 |  |
| void | void | 设置当前地图为不活跃状态 | 地图不活跃即保存地图 |
| void | bool | 获取当前地图的状态 | 状态：活跃、不活跃 |
| void | void | 设置地图bad |  |
| void | bool | 获取地图是为bad |  |
| void | void | 清空地图 | 用于地图重置 |
| set<GeometricCamera\*> | void | 地图预保存 | 用到的相机类型 |
| KeyFrameDatabase\*；  ORBVocabulary\*； map<unsigned int, GeometricCamera\*>; | void | 地图后加载 | 参数分别为：  共用的关键帧数据库  共用的字典  地图用到的相机类型 |
|  |  |  |  |
| longunsigned int | KeyFrame\* | GetKeyFrame()取地图中具体关键帧 |  |
| longunsigned int | MapPoint\* | GetMapPoint()获取地图中具体地图点 |  |
| KeyFrameBlock\* | void | AddKeyFrameFromKFBlock()将KFB块中的KF添加到map中 |  |
| MapPointBlock\* | void | AddMapPointFromMPBlock()将MPB块中的MP添加到map中 |  |
| vector<KeyFrame\*> | void | DeleteKeyFrames()删除已经存储的关键帧 |  |
| vector<MapPoint\*> | void | DeleteMapPoints()删除已经存储的地图点 |  |
| unsigned long int | void | SetDelKFID()设置关键帧删除界线 |  |
| unsigned long int | void | SetDelMPID暂时未用 |  |
| unsigned long int | void | GetDelKFID()获取关键帧删除界线 |  |
| unsigned long int | void | GetDelMPID()暂时未用 |  |