

LayeredCostmap类的地图更新函数主要分为两步，先更新bound，再更新cost，它调用Layer类方法，它在各层子地图中被重载。CostmapLayer类作为静态层和障碍层的基类，提供了一些对地图层进行操作的函数。

#### 1.Costmap2DROS

#### 构造函数 Costmap2DROS::Costmap2DROS

首先是一些参数的获取。循环等待直到获得机器人底盘坐标系和global系之间的坐标转换。  
并获取rolling\_window、track\_unknown\_space、always\_send\_full\_costmap的参数，默认为false。

#### 地图更新线程 Costmap2DROS::mapUpdateLoop

这个函数循环调用UpdateMap函数，更新地图。并以publish\_cycle为周期，发布更新后的地图。

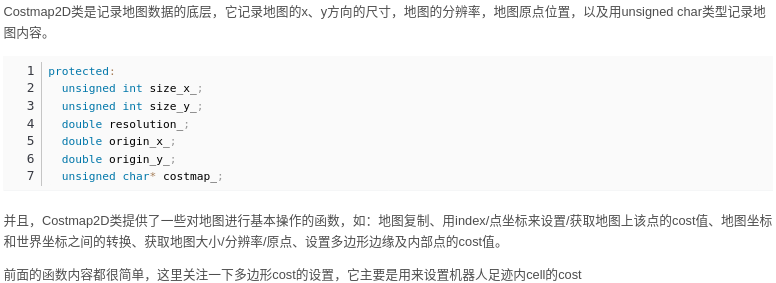
#### 地图更新 Costmap2DROS::updateMap()

这个函数首先调用类内getRobotPose函数，通过tf转换，将机器人底盘系的坐标转换到global系，得到机器人的位姿。然后通过layered\_costmap\_调用LayeredCostmap类的updateMap函数，更新地图。

#### 激活各层地图 Costmap2DROS::start()

start函数在Movebase中被调用，这个函数对各层地图插件调用activate函数，激活各层地图。

#### Costmap2D类



#### 设置多边形cost Costmap2D::setConvexPolygonCost

先将世界系下的多边形顶点转换到地图坐标系，并存放进map\_polygon数组中。

#### 获取多边形边缘及内部cell Costmap2D::convexFillCells

首先确保给定的多边形顶点不少于3个，接着调用类内polygonOutlineCells函数，通过给定的顶点提取多边形边上的cell。

#### 获取多边形边上的cell Costmap2D::polygonOutlineCells

这个函数循环调用raytraceLine函数，不断获取相邻之间的连线，最终组成多边形边上的cell，需要注意的是需要将最后一点和第一点连接起来，形成闭合。

#### 两点连线上的cell raytraceLine

对于离散的平面点，指定两个点，这个函数可以找到两个点之间的其他点，使得这些中间组成一个尽可能趋近直线的点集。

#### LayeredCostmap类

LayeredCostmap类是Costmap2DROS的成员，含有主地图，并能通过它操作各层子地图。这里关注一下两个被Costmap2DROS调用的函数。

#### 地图尺寸设置 LayeredCostmap::resizeMap

这个函数在Costmap2DROS动态配置参数的回调函数ReconfigureCB中被调用，作用是在开启地图更新线程之前，调用Costmap2D的resizeMap函数，用给定参数重新设置主地图的尺寸、原点、分辨率，再通过plugin指针调用各层地图的matchSize，使其以上参数和主地图匹配。

#### 地图更新 LayeredCostmap::updateMap

这个函数在Costmap2DROS的地图更新线程中被循环调用。它分为两步：第一步：更新bound，即确定地图更新的范围；第二步：更新cost，更新每层地图cell对应的cost值后整合到主地图上。

#### CostmapLayer类

这个类继承自Layer类和Costmap2D类，它是地图插件（静态层和障碍层）的基类。它的类方法主要用于处理bound和用几种不同的策略合并子地图和主地图。

