障碍层地图通过订阅传感器话题，将传感器输出的障碍物信息存进buffer（剔除过高、过远的点），在本层地图上将观测到的点云标记为障碍物，将传感器到点云点连线上的点标记为FREE\_SPACE。最后，在bound范围内，将本层地图合并到主地图上。



#### 初始化 StaticLayer::onInitialize

首先从参数服务器加载参数，确定rolling\_window\_、track\_unknown\_space的值，并调用matchSize函数，根据主地图的参数来设置障碍层地图。

#### 回调函数 ObstacleLayer::laserScanCallback

这里以LaserScan数据的回调函数为例，先将收到的message的laser数据转换为sensor\_msgs::PointCloud2格式的数据，再调用ObservationBuffer类的bufferCloud函数，将点云数据存到buffer中。

#### 更新障碍地图边界 ObstacleLayer::updateBoundsFREE

这个函数主要完成：clearing、marking以及确定bound。

和静态地图类似，同样也是先判断是否是rolling地图，若是则更新地图原点。

#### 清理传感器到障碍物间的cell ObstacleLayer::raytraceFreespace

先将传感器坐标转换到地图坐标系。

#### 更新障碍地图代价 ObstacleLayer::updateCosts

这个函数就是将机器人足迹范围内设置为FREE\_SPACE，并且在bound范围内将本层障碍地图的内容合并到主地图上。