# ScanMatcher::score

1. 计算出激光传感器在世界坐标系下的坐标
2. 针对不同的beam，计算出其障碍物在世界坐标系下的位置=phit，

对于障碍物周围的8个grid进行筛选判断这个grid是否为不透明且其沿着光束方向上的grid是否为透明的？如果结果都为是则计算mu=phit-cell.mean，在8个grid中找到最小的那个mu值作为此条beam的bestmu值。



1. 得到所有beam的bestmu后，根据如下的公式计算此刻激光测量的可信度：



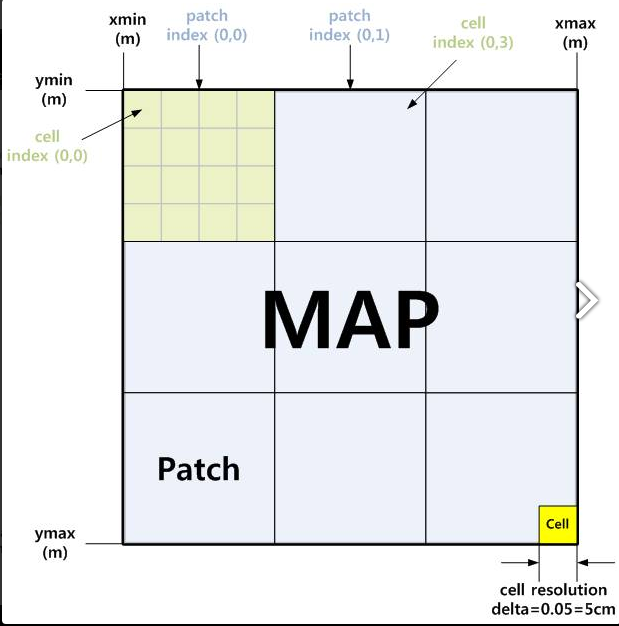
# 二、地图的存储方式

在ScanMatcher::score函数中根据grid地图坐标表示取出对应的grid的信息：

const PointAccumulator& cell=map.cell(pr);

首先这里我们先讲解地图的存储方式：





根据我们对地图的大小设定，对于grid地图来说大小为2000\*2000，这2000\*2000的地图又被分为大小均等的patch，其patch大小为32\*32。而每个patch中又由32\*32个grid组成。整张地图对应的数据结构为HierarchicalArray2D<PointAccumulator>，每一个patch对应的数据结构为Array2D<PointAccumulator>，每一个grid对应的数据结构为PointAccumulator，而HierarchicalArray2D继承了Array2D<autoptr< Array2D< PointAccumulator > > >，Array2D类中拥有最重要的一个成员变量m\_cells。下面我们解释HierarchicalArray2D中的m\_cells和Array2D中的m\_cells区别：

HierarchicalArray2D中的m\_cells存储的指针指向Array2D<PointAccumulator>；

Array2D< PointAccumulator >中的m\_cells存储的指针指向PointAccumulator；

在我们已经知道grid坐标系下的坐标后如何提取出对应grid的信息呢：

首先我们要求出patch对应的坐标：patch坐标=grid坐标/32

然后再在patch中求出对应的grid坐标：patch中的坐标=grid坐标的后五位

# 三、ScanMatcher::optimize



根据函数输入参数得到的激光传感器在世界坐标系下的位姿，调整其位姿态分别计算其上下左右相距初始位置0.05米的测量置信度分数，并取其对应最大置信度分数的位姿作为修正后的激光传感器的位姿。如果上下左右的四个位置的置信度分数相同，那么就取0.05\*0.5米作为相距初始激光传感器位置的距离，这种精细划分不能超过5次。函数返回的是修正过后的激光传感器的位姿和其对应的置信度分数。

# 四、GridSlamProcessor::propagateWeights



假设我们地图中有三个粒子，其权重分别为10、20、30。这三个粒子中的节点结构如图中所示拥有D、E和F父节点。在调用propagateWeights之前父节点D、E和F的accweight为0，调用propagateWeights函数之后结果如图：



# 五、GridSlamProcessor::resample

resample函数所做的工作是：根据粒子的权重选择复制某些粒子放弃权重小的粒子，生成新的粒子簇。如上图所示我们将粒子2和粒子3进行了复制，但是粒子中的节点信息不是完全的复制，我们需要设置t时刻粒子1、2节点的父节点为t-1时刻粒子2的节点；t时刻粒子3、4、5的父节点为t-1时刻粒子3的节点。对于被复制的粒子2和粒子3，将其权重设置为0。对于被删除的粒子1、粒子4和粒子5，需要清除其在GridSlamProcessor ::m\_particles数据结构中节点的信息。

清空GridSlamProcessor ::m\_particles所有粒子，调用registerScan函数修正粒子对应的地图，最后将复制得到的粒子压入到GridSlamProcessor ::m\_particles中。

如果不满足重采样条件，如下图

只是改变了t时刻粒子中节点里面的父节点的信息，同时领t时刻的节点中的reading=0；

# 六、地图膨胀（map::resize）

在computeActiveArea函数中有对地图膨胀的相关代码，主要方法如下：



原有地图大小是蓝色框表示，t时刻在障碍物patch中检测到有障碍物，或者也可以理解为这个patch被激光扫描到了。则更新地图的大小(红色框表示)、地图的坐标系。

# 七、drawFromMotion

gmapping使用的是里程计模型(见书probabilistic robotics132页)，有几个比较重要的函数需要首先详细说明一下：

**absoluteDifference(p1，p2)**：p1是t时刻里程计小车位置，p2是t-1时刻里程计小车位置，得到的结果是p1在p2坐标系下的描述。



**absoluteSum(p,noisypoint)：**p是t-1时刻小车的位置，noisypoint是p1在p2坐标系下的坐标叠加上噪声的结果。



首先将noisypoint在p1坐标系下的坐标转换成世界坐标系下的坐标result，再将p点的坐标与result相加得到最终的位姿点。