# MonoLaneMapping

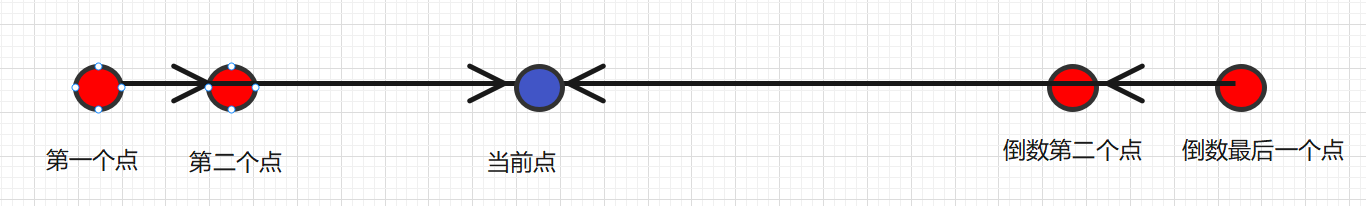
## 1、当前帧车道线每个点的噪声计算

其中，x和y分别是当前帧某车道线的坐标，剩下的其余参数都是人为设定的。

## 2、当前帧车道线计算每个点的阈值（get\_dist\_thd函数）

其中，除了xyz变量是根据当前帧拿到的，其余的变量全部是人为设定的。

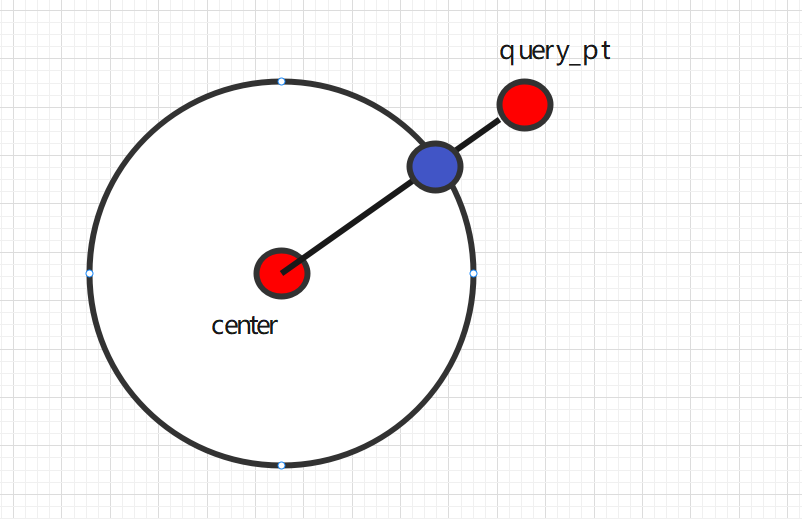
## 3、get\_pts\_to\_add



得到一条车道线上的所有点，计算第一个点和第二个点的向量n1，倒数第二个点和倒数最后一个点的向量n2，遍历所有的点，对于所有的点计算第一个点和最后一个点的向量n\_start和n\_end，分别计算n\_start和n1向量的角度，n\_end和n2向量的角度，如果任何一个角度小于阈值，就将这个点加入到要返回点中。这个个函数的作应该是滤除杂点。

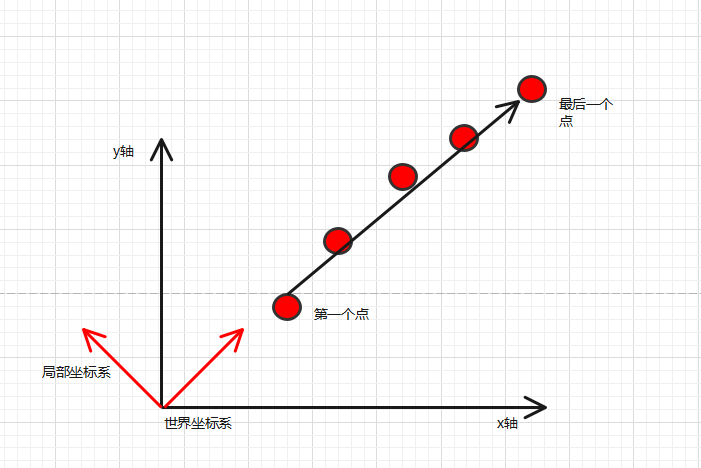
## 4、get\_nearest\_on\_circle

这个函数的根本目的是给定 圆的center和半径R和一个查询点，在圆上找到一个距离这个查询点最近：



基本方法是计算得到center到query\_pt构成的向量，并将这个向量单位化乘以半径R后加上center即可得到结果蓝色点。

**5、fitting**



先和地图中匹配上的当前帧车道线变换到世界坐标系下，如上图中的红色点，然后根据第一个点和最后一个点构成的向量与世界坐标系下的x轴计算角度，并构建局部坐标系。并将所有在世界坐标系下的点变换到局部坐标系下。

使用局部坐标系下所有点的x和y拟合得到x-y三次多项式线，x和z拟合得到x-z三次多项式线。

## 6、get\_skeleton(init\_ctrl\_pts函数走向)

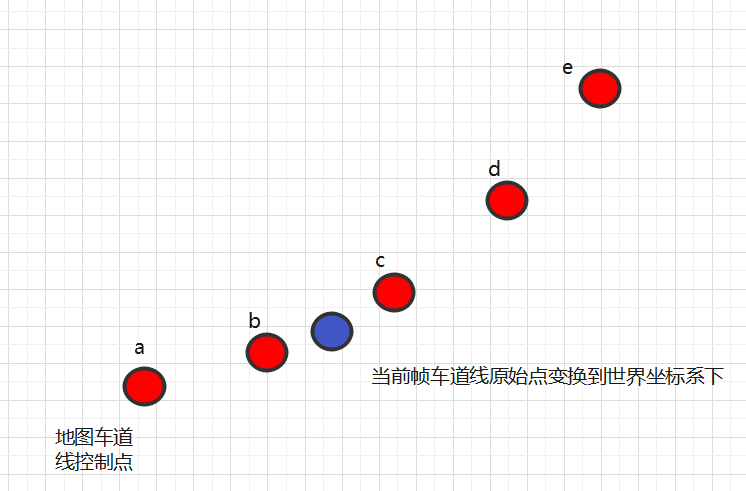
输入的是当前帧某条车道线的原始点和拟合过的多项式曲线表达式：

## 7、 get\_skeleton(update\_ctrl\_pts函数走向)

## 8、find\_footpoint

当前帧感知结果的车道线和地图中的车道线已经匹配，如何找到当前帧车道线每个点对应的样条曲线的u。

作者是这么实现的：假设当前帧车道线和地图中的车道线已经匹配；然后找到哪两个地图中车道线的控制点和当前帧的点最近。



如上图中，蓝色点为当前帧点，红色为地图车道线控制点。找到最近的两个控制点为b和c。使用a-d四个控制点构建CatmullRomSpline曲线，这个样条曲线的特点是a-d构建的样条曲线，作用域只在b到c。

下一步是计算b-c作用域对应的u参数。我们知道作用域的起点是b，结束点是c。则：

则u = 0.25 + ratio \* (0.5 – 0.25)