# 地磁指纹建库

## 0、算法基本流程

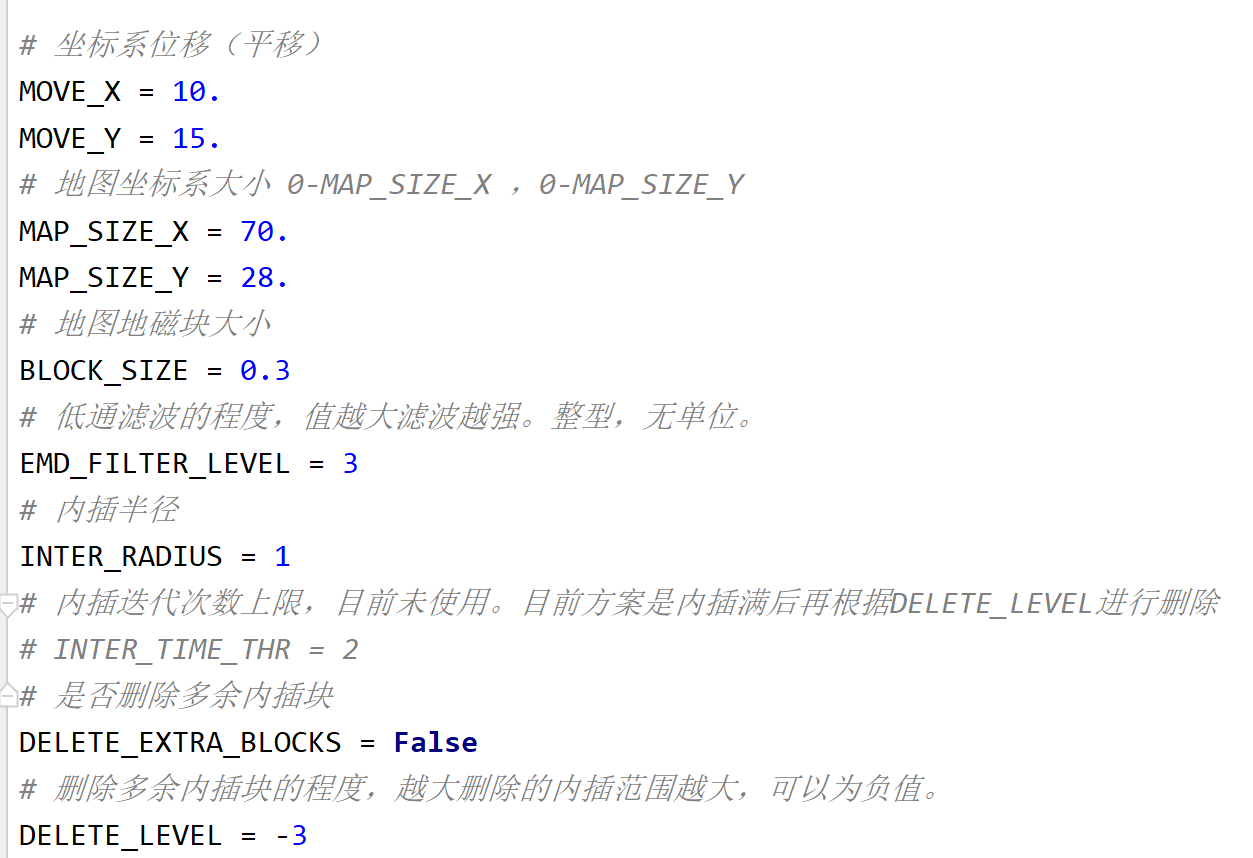
·使用的输入文件是：带iLocator坐标的手机IMU数据文件（二者对齐合并）。

·建库函数原理：

1. 读取地磁mag(TYPE\_MAGNETIC\_FIELD)、方向四元数quat(TYPE\_GAME\_ROTATION\_VECTOR)、iLocator坐标。
2. 将地磁变换为垂直、水平分量，且取绝对值保证为正数；
3. 滤波、栅格化、内插填补；
4. 输出结果，得到两分量平行指纹库文件，mv.csv mh.csv。

## 1、设置地图参数

**这些参数目前基本为经验值，只需要根据实际效果微调。**



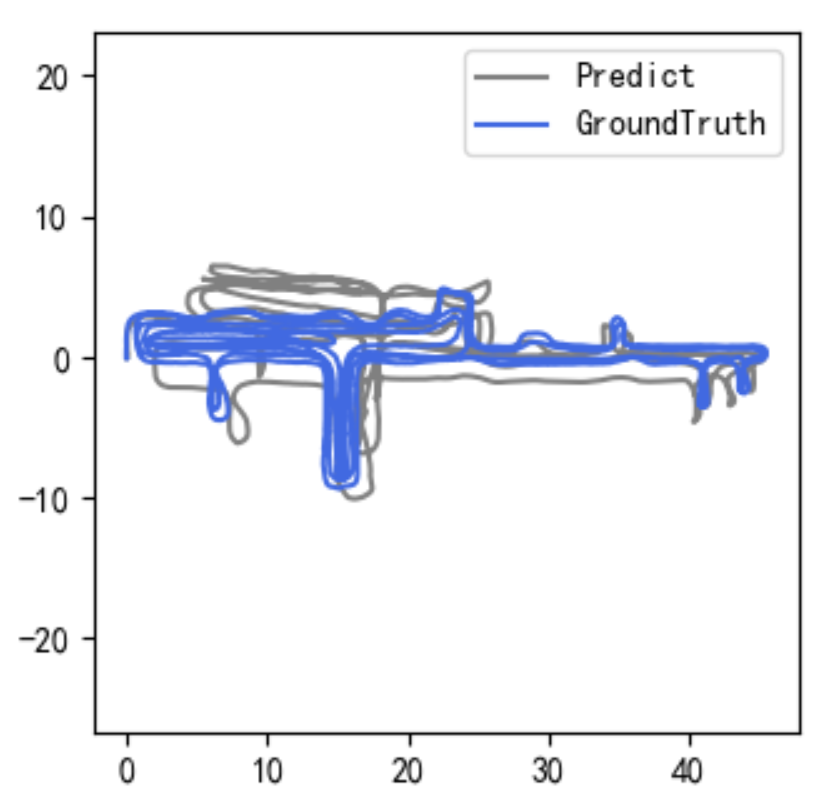
### ·MOVE\_X, MOVE\_Y：单位（米）

iLocator坐标系正数化，平移iLocator坐标系，保证所有坐标均为正数（最好预留边缘一些）：

比方说这个是我在星湖楼8楼采的建库轨迹**之一**。（蓝色的线是iLocator的坐标轨迹，灰色的是PDR，当然建库不需要PDR坐标，坐标轴单位m。）

此时iLcaotor打点的起点1P1是(0,0)，可以看到iLocator的轨迹并不全在**第一象限**。然后我就设置平移MOVE\_X=10，MOVE\_Y=15，保证所有坐标落在第一象限。

而iLocator打点固定的话，不同iLocator轨迹之间的坐标系是一样的，所以使用同一个MOVE\_XY即可。



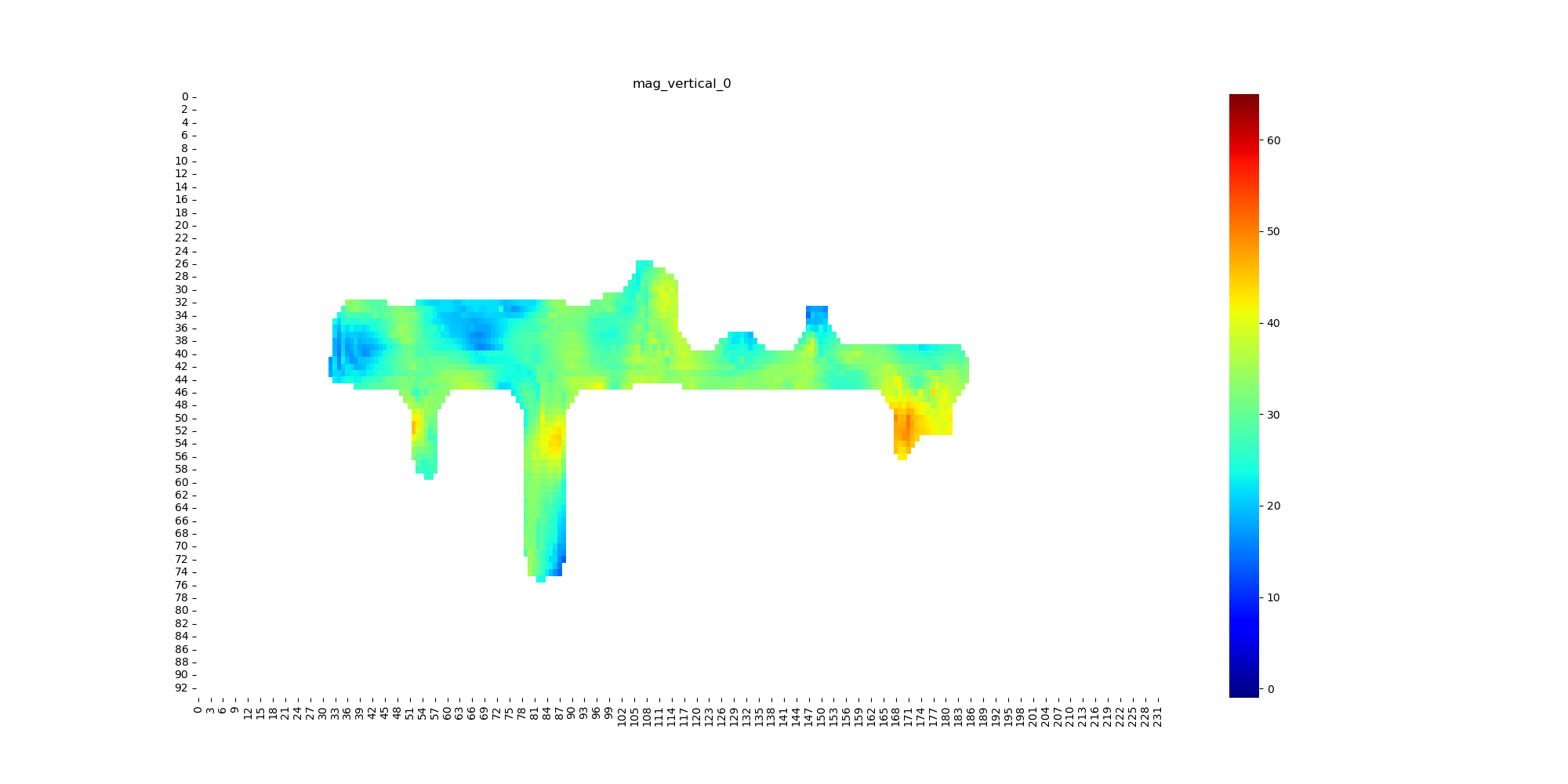
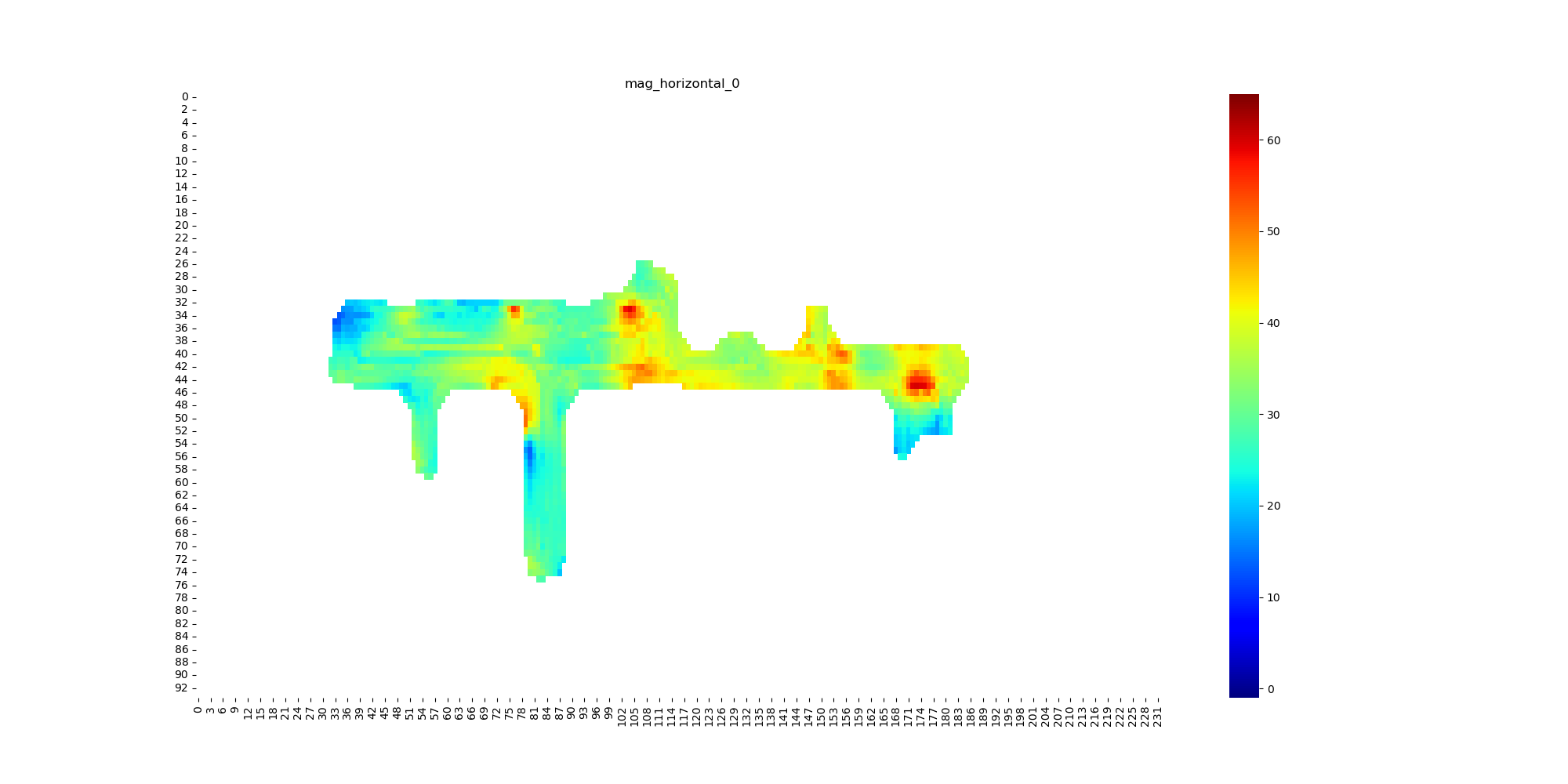
### ·MAP\_SIZE\_X，MAP\_SIZE\_Y：单位（米）

指纹库大小，强制要求：

MAP\_SIZE\_X > 平移到第一象限的建库坐标最大x值；

MAP\_SZIE\_Y > 平移到第一象限的健康坐标最大y值。

可以预留一些边缘，这是最终的星湖楼8楼建立的库，如下图，此时坐标是指纹库二维数组的下标（MAP\_SIZE / BLOCK\_SIZE）。



### ·BLOCK\_SIZE: 单位（m）

栅格化时的块大小，一般场景大的就0.3m，场景小的就0.25m。所以地磁指纹二维数组的大小就是[ MAP\_SIZE\_X / BLOCK\_SIZE ]\*[ MAP\_SIZE\_X / BLOCK\_SIZE]。

### ·EMD\_FILTER\_LEVEL：

滤波参数，不用管

### ·INTER\_RADIUS：单位（m）

内插填补不存在的地磁块时，参考的范围，不用怎么改

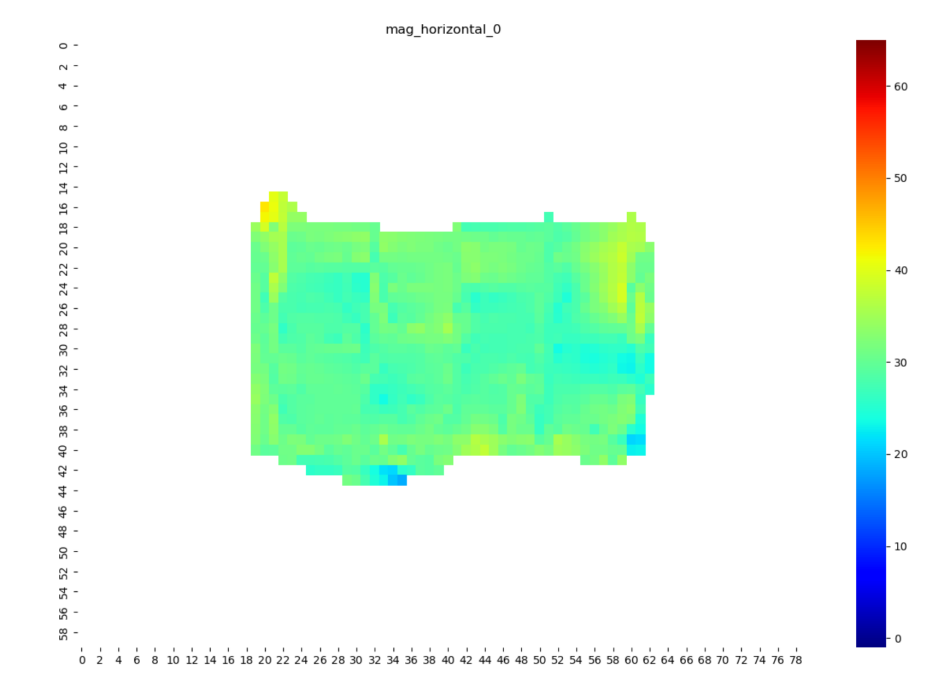
### ·DELETE\_EXTRA\_BLOCKS：

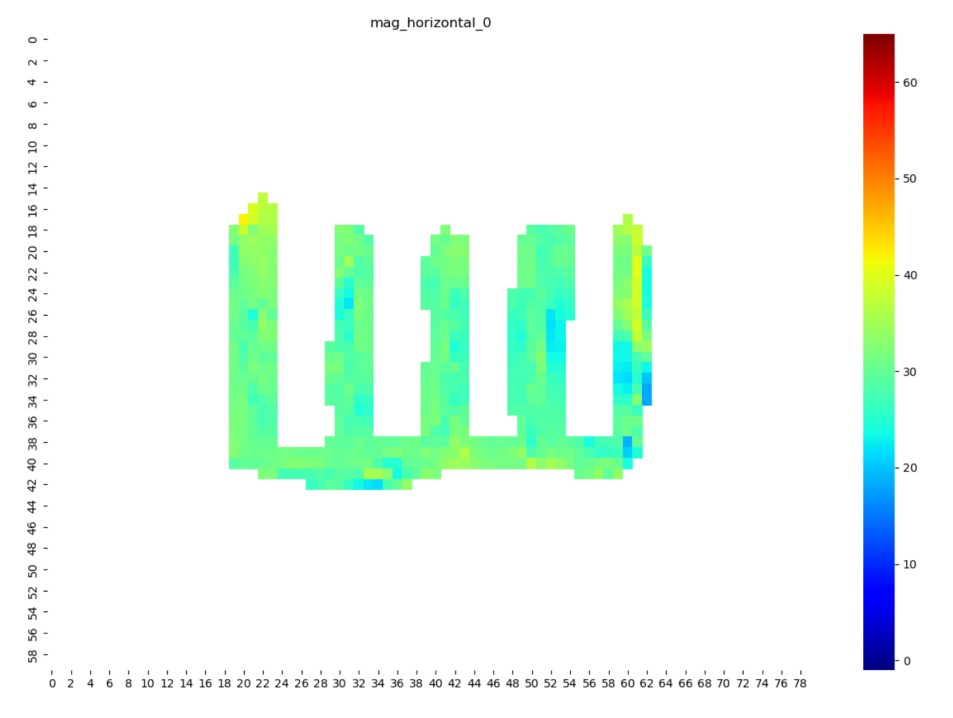
是否删除内插填补多余的块。在内插填补时，会填补尽可能所有的空格，但这可能会使地图上本来是障碍物的地方也填上了指纹，增加错误的可能。

### ·DELETE\_LEVEL：

删除多余内插块的程度，越大删除的内插范围越大，可以为负值，只在DELETE\_EXTRA\_BLOCKS = True的时候生效。当=0的时候，只会保留INTER\_RADIUS长度范围内的内插块，<0的时候会保留 INTER\_RADIUS + abs(DELETE\_LEVEL)范围内的内插块。

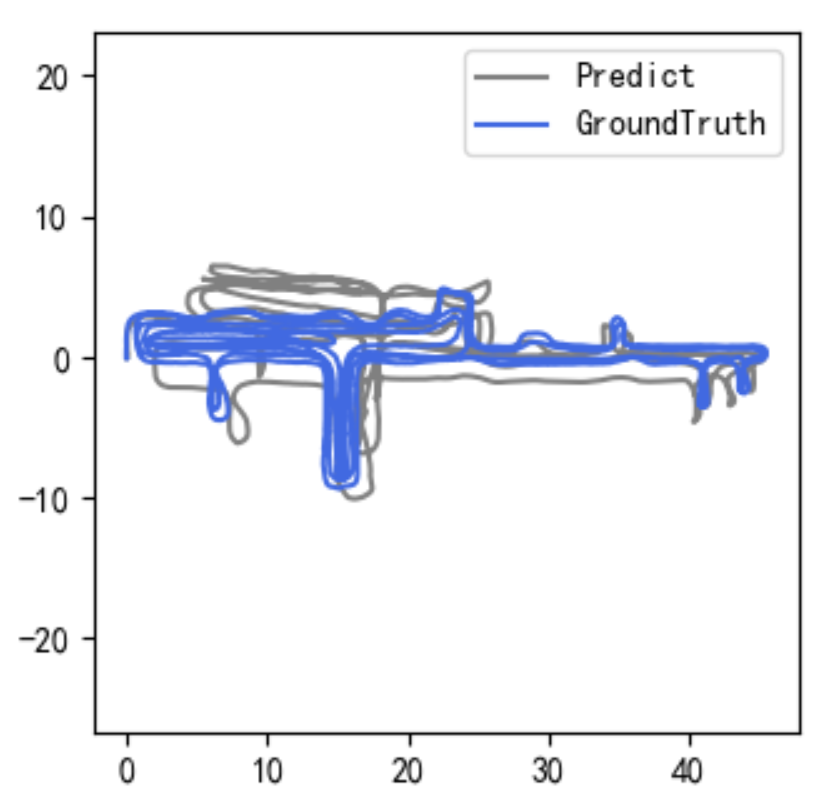
删除多余插值块的前后对比图（机房中间的一排排服务器）：



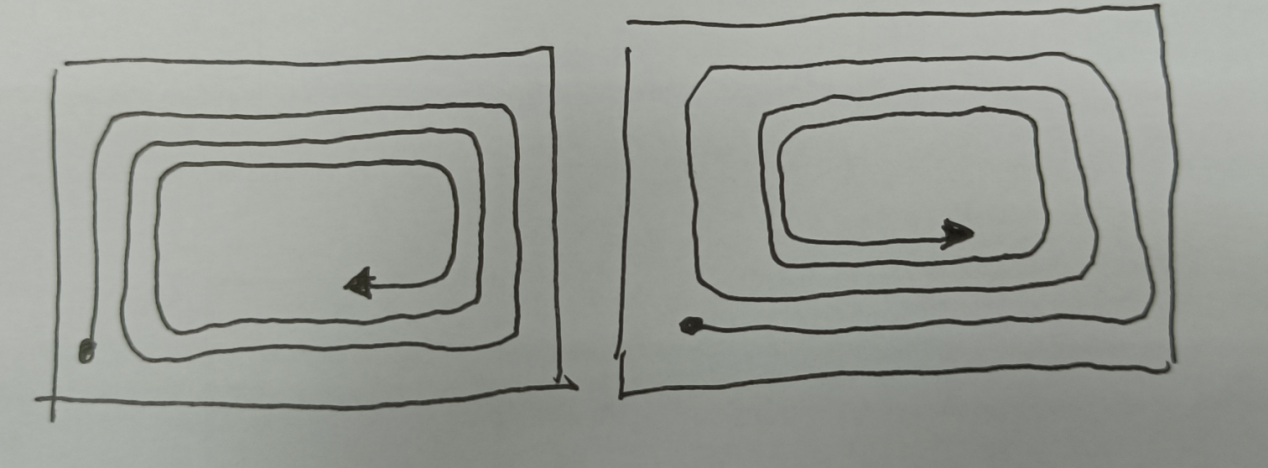


## 建库轨迹要求

·遍历覆盖地图，轨迹之间间隔不超过1m，越密越好。比方说这个是我在星湖楼8楼采的建库轨迹**之一**。（蓝色的线是iLocator的坐标轨迹，灰色的是PDR，当然建库不需要PDR坐标，坐标轴单位m。）



·一般一张地图，我是**顺时针全覆盖走一趟+逆时针全覆盖走一趟**，拿这两文件一起建库。而且我是螺旋着走，不是S型走。



# 地磁匹配

PDR + 地磁指纹 + 枚举粗搜索 + 高斯牛顿迭代法局部优化

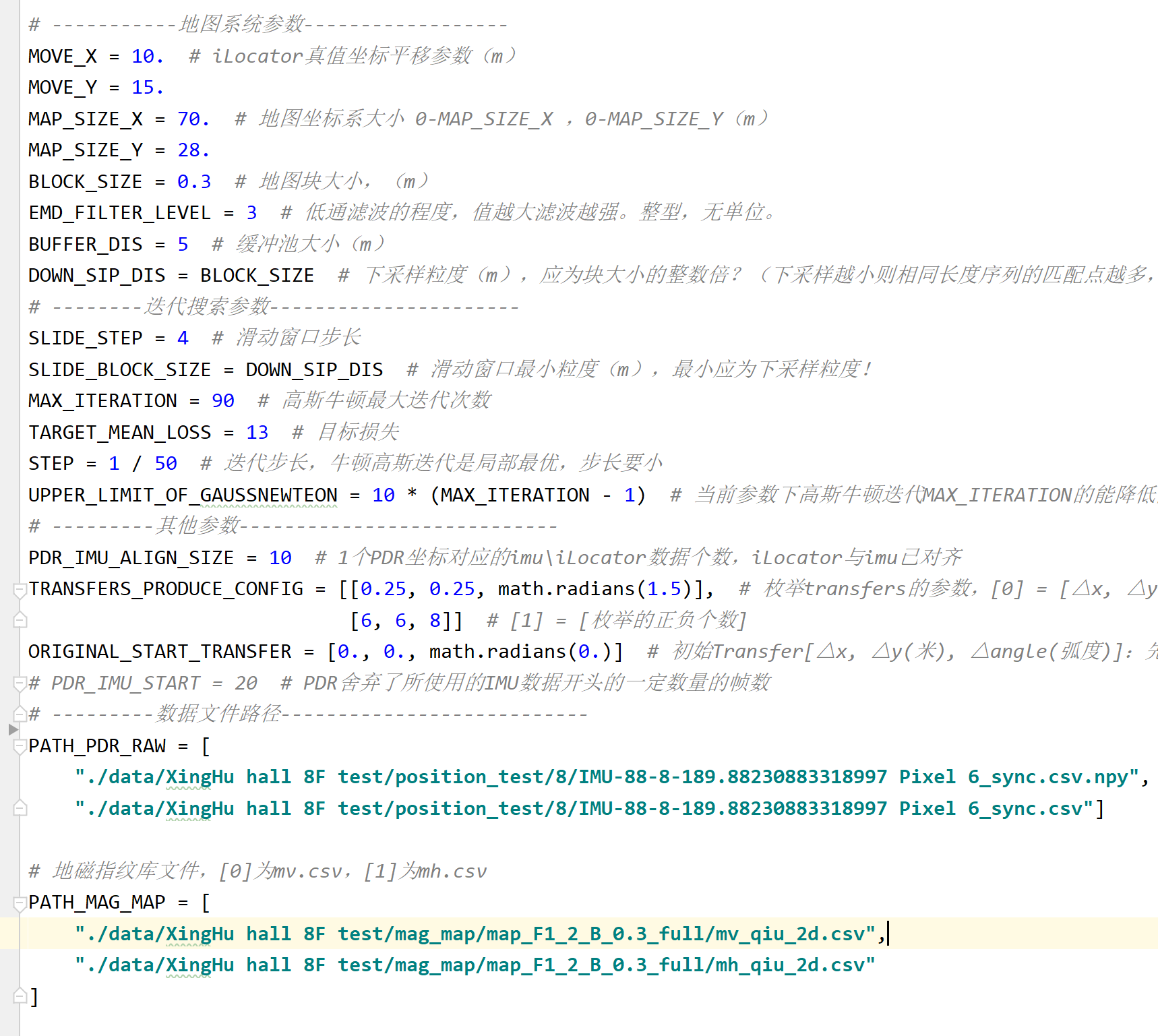
## 约束条件

保证PDR的第一段是准确的（即认为PDR的起始段已经准确地定位到磁指纹地图中），否则如果第一段都枚举、匹配失败，后面就也全都枚举不到正确的。因为现在的枚举范围已经很小了，但是时间复杂度已经很高了。如果无法提供准确的第一段PDR，只能全图枚举匹配搜索，而且就算这样也会出现好几段候选起点PDR段。

我们的PDR和iLocator是按起始段对齐的，所以不需要设置ORIGINAL\_START\_TRANSFER。

## １、设置匹配参数

**这些参数目前基本为经验值，只需要根据实际效果微调。**



·MOVE\_X\Y:

与建库同理

·MAP\_SIZE\_X\Y：

与建库同理，匹配并不需要这个参数，这里只用来绘图。

·BLOCKE\_SIZE：

与建库同理

### ·BUFFER\_DIS：单位（米）

滑动窗口大小（地磁匹配序列长度），本程序使用距离，而非帧数。

### ·DOWN\_SIP\_DIS：单位（米）

下采样粒度，PDR输出的坐标频率太高了，导致一段BUFFER\_DIS的轨迹中的匹配点数太多。

### ·SLIDE\_STEP：正整数

滑动窗口滑动距离 = SLIDE\_BLOCK\_SIZE \* BLOCKE\_SIZE

### ·SLIDE\_BLOCK\_SIZE

滑动窗口滑动距离最小粒度， >= DOWN\_SIP\_DIS

### ·MAX\_ITERATION

匹配搜索时，高斯牛顿迭代法最大循环次数

### ·TARGET\_MEAN\_LOSS

认为匹配成功的地磁距离阈值，匹配段与指纹库候选段的磁指纹欧氏距离。

### ·STEP

高斯牛顿迭代法，迭代步长。

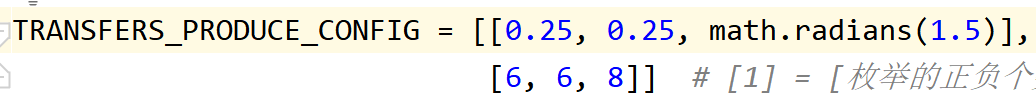
### ·UPPER\_LIMIT\_OF\_GAUSSNEWTEON

手动设置的，当前参数下高斯牛顿迭代MAX\_ITERATION的能降低的loss上限。用来剪枝，避免浪费搜索时间在不可能成功的匹配段。

### ·PDR\_IMU\_ALIGN\_SIZE

主要要解决的是PDR坐标和手机IMU数据的对齐，保证**赋予PDR(x, y)其对应的手机IMU数据**。比方说我们的PDR是滑动10个IMU数据输出1个PDR坐标，开头没有切掉IMU数据，则PDRxy.mag[i] = IMU.mag[(i+1)\*10] ... ...

### ·TRANSFERS\_PRODUCE\_CONFIG：单位（米、弧度）



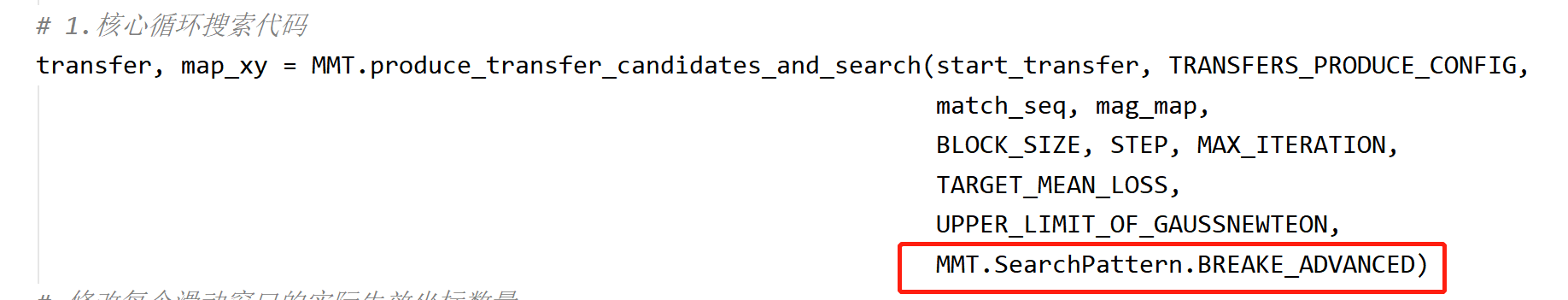
1. = [△x, △y(米), △angle(弧度)]
2. = [枚举的正负个数]

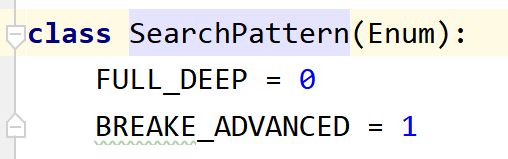
这代表枚举搜索的时候，三重for循环，按这个粒度与个数进行搜索。这个例子会在范围 X ± 0.25 \* 6 米， Y ± 0.25 \* 6米，角度±1.5 \* 8 度，的范围内，从近到远。

### ·ORIGINAL\_START\_TRANSFER：

初始变换向量，Transfer[△x, △y(米), △angle(弧度)]：先绕原坐标原点逆时针旋转，然后再平移。这个量只是用来协助达到PDR**约束条件**的，让PDR初始段是与真值一致的。

### ·SearchPattern.BREAKE\_ADVANCED：





两种搜索模式：

FULL\_DEEP：计算所有枚举候选项的loss与结果，最后保留min loss对应的map\_xy；缺点：无剪枝，时间复杂度高。

BREAKE\_ADVANCED：当枚举过程中，遇到loss < target loss，直接返回，不继续搜索更小的loss。