

RadarCAS 说明

此项目用于实时接收以及显示雷达和GPS信息，并累计到一定数量以后计算特征，在第二次测量时，将特征和第一次测量时的特征做对比，并输出对应的GPS信息。

项目结构

- RadarCAS 项目
 - res: 项目所用到的图标文件
 - connexions: 连接GPS以及雷达的类文件
 - value: 语言本地化文件strings，以及缺省文件路径定义：respath
 - test: 单元测试文件，主要用于测试项目内不在类里的函数
 - tensorflow: 神经网络core代码
 - nets: 神经网络拓展部分
 - logs: 暂时没用上，如果把appconfig中logging设定为非DEBUG模式则logs会输出到此文件夹内
 - utils: 文件解析，格式转换等函数
 - scratchMesh: 请忽略这个文件夹，它是草稿纸
 - data: 默认存放数据的文件夹
 - mocks: 默认存放模拟数据的文件夹
 - performance: 存放性能测试数据文件夹
 - *.py: 此项目的图形界面以及数据控制的代码

使用

工具栏

鼠标放到按钮上，有些有提示信息。



- 开始: 开始采集按钮
- 停止: 停止采集按钮
- 雷达配置: 配置雷达以及算法相关参数
- GPS配置: 配置GPS串口参数
- 测量论设置: 设置测量轮相关参数
- 更新频率: 在不同采集模式下的线程相关频率设置
- 保存路径: 数据存放地
- 融合GPR: 一个将GPS以及雷达采集来的数据融合为GPR格式的小工具

程序可以以4种模式进行测试:

1. 使用模拟GPS数据，模拟雷达数据，当数据读取完后自动停止
2. 使用模拟GPS数据，真实雷达数据，当用户按了停止或者模式数据读取完了以后自动停止
3. 使用真实GPS数据，模拟雷达数据，当用户按了停止或者模式数据读取完了以后自动停止

4. 使用真实GPS数据， 真实雷达数据， 当用户按下停止或者出现不可预知的错误自动停止采集

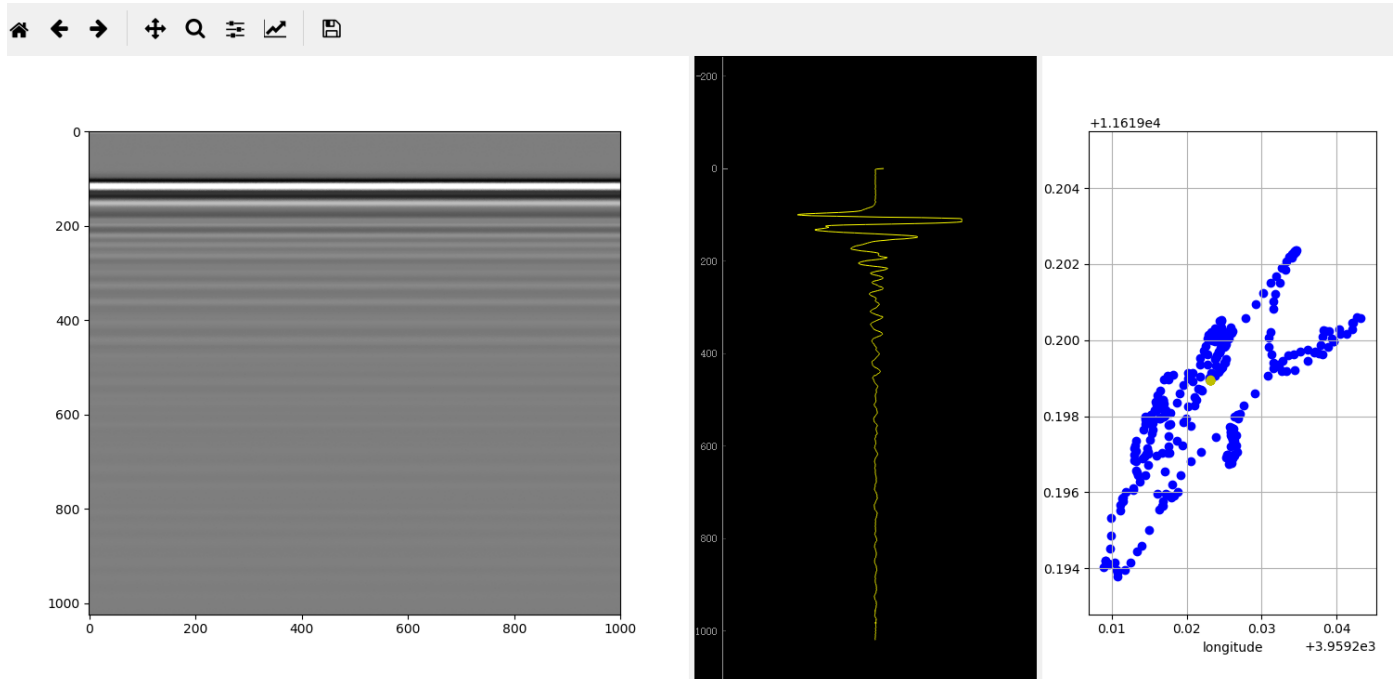
每种模式经过测试，至少可以接受2W数据以上，最高测试量是10W，由于等待时间太长，只尝试了一种模式：

- 2021/1/22 10:03=====25674-23325 data with mode: [Mock GPS, Mock Radar] <PASS>
- 2021/1/21 10:12=====25682-23325 data with mode: [GPS, Mock Radar] <PASS>
- 2021/1/21 10:28=====25666-37000 data with mode: [Mock GPS, Radar] <PASS>
- 2021/1/21 11:34=====30411-31800 data with mode: [GPS, Radar] <PASS>
- 2021/1/25 13:34=====25674-23325 data with mode: [Mock GPS, Mock Radar] <PASS>
- 2021/1/25 10:12=====25682-23220 data with mode: [GPS, Mock Radar] <PASS>
- 2021/1/25 09:28=====48200-34510 data with mode: [Mock GPS, Radar] <PASS>
- 2021/1/25 13:34=====34000-2888 data with mode: [GPS, Radar] <PASS> 但是由于Radar在接收3W左右信息时会出现python无法立即完成一个非阻止性套接字操作的错误。。自动断开了，有可能是接收太快导致的吧？
- 2021/1/25 14:28=====54853-51965 data with mode: [GPS, Radar] <PASS> 果然是因为太快导致的。。想不到我的程序也有速度超过雷达的一天：），已经修改好了
- **2121/1/26 12:45=====100227-97712 data with mode[GPS, Radar] <PASS>** 10W条数据收集通过，但是到后面会出现每300条左右有些卡顿的情况，估计是因为画GPS点图引起的，不过问题应该不大。

注意，在第一次测量前勾选“使用GPS”，但第一次测量结束后会自动取消勾选，因为第二次测量不需要使用。

图

- 左边图为最新一段雷达数据的灰度图，更新速度较慢
- 中间为当前最新采集的波形图，更新速度最快
- 右边为GPS采集的数据，更新速度比较慢



状态栏

状态栏处于最底部：

数据收集计数器: 0 当前特征计算索引: 416 Prior数据长度: 0 Unregistered数据长度: 0 prior移动距离: 0.0m Unreg移动距离:

- 数据收集计数器：显示当前收集到的数据长度
- 当前特征计算索引：显示下一个所需要计算的特征索引，由于计算速度比收集速度慢，当用户点击停止按钮后，特征计算尚未完成，让用户得知特征计算的进度
- Prior数据长度：第一次测量时收集的数据长度
- Unregistered数据长度：第二次测量时收集的数据长度

如果用户使用测量轮模式，状态栏还会多显示两个距离标签：

数据收集计数器:	0	当前特征计算索引:	416	Prior数据长度:	0	Unregistered数据长度:	0	prior移动距离:	0.0m	Unreg移动距离:	
----------	---	-----------	-----	------------	---	-------------------	---	------------	------	------------	--

- Prior移动距离：第一次测量所走过的距离
- Unreg移动距离：第二次测量所移动的距离

可能存在的问题

1. 计算线程停止以后依然会运行一段时间

从下面这一段日志可以看到，明明已经停止了，但是似乎依然运行了一段时间，由于需要融合数据并保存，这可能会导致卡死，暂时没找到原因，似乎更早的版本并没有出现这个问题。

可能的原因： PYTHON的线程有可能是一种回溯的运行方式，在线程停止后，依然会把先前在内存中尚未循环完的任务完成，也就是这段日志中出现的Start calculate==>Counter : 0 Current calculate index: 416（当然这一段任务运行速度非常快，只是如果被堵塞卡在线程中），而之前版本是按照pickle格式保存，不存在数据融合的步骤，速度快很多而不至于堵塞，让人感觉不出来线程的回溯。

```
01/22/2021 09:37:28 AM - INFO - mainFrame.py - start_calculate_action - 535 - Prior Calculate done..
GPR DATA LENGTH : 54896353
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
Start calculate==>Counter : 0 Current calculate index: 416
01/22/2021 09:38:36 AM - INFO - mainFrame.py - before_start_collection - 391 - Before starting thread...
```

可能的解决方法： 把数据融合为GPR的步骤分散到收集数据的步骤中，每收到N条融合一次，尚未试过，担心会导致收取数据上时间的问题。目前也不太确定是否是这个融合过程慢（2.5W条需要4s左右）而导致的线程无法及时停止。

2. Radar或者GPS断开连接的处理

如果在测量的过程中，雷达或者GPS断开连接，或者GPS发送空数据，可能会导致程序崩溃。

- 目前雷达如果检测到连接断开，会自动重连，但是由于重连到雷达速度奇慢无比，而且似乎从未重连成功，这会导致之后的线程堵塞从而主线程崩溃。目前的处理是，如果radar突然断开连接，则自动停止测量。
- 如果GPS突然断开，一般有两个可能性：
 - 如果从电脑处拔出蓝牙设备，会有设备已经被删除的错误提示
 - 如果蓝牙设备自动关机或者过远，会一直接到全空字节，程序计算连续超过N个空字节则判断已经和GPS断开连接
 GPS断开处理方式也和Radar处理方式相同，自动停止数据收集。
- 但是如果GPS定位失效收到了空的三维数据，暂时不知道如何处理，目前就直接复制上一个数据

3. 真实数据下3.7W数据程序崩溃 此条已经修改好

在同时使用GPS和雷达收数据的模式下，第一次测量，在数据量到3.7W条时崩溃了。正在尝试修改radar collector线程。

4. 雷达电量低时会出现不可预知的错误

在26号早上到下午测试10W数据量之后，也许雷达电压过低，发送数据异常，导致算法中格式转换出了差错，但由于过后雷达没电关闭了，无法再复现这个错误，尚未能查明究竟是数据的什么问题导致的。稍微更具体点：

```
# findWayToHome.py - 112 line
singleWindowRadarData = singleWindowRadarData[self.firstCutRow:self.firstCutRow + self.patchSize, :]
```

异常数据在这一行会出现IndexError: too many index in array的错误，再往前追踪，可以查出是

```
# findWayToHome.py - 110 line
singleWindowRadarData = np.array(self.radarData[headIndex:headIndex + self.patchSize]).T
```

这一行出了问题，原始数据转成Numpy格式的shape并非所期待的(416,1024)，而是 (416[1024],) 并且每个element是list并且有1024个数字。具体为啥会这样，尚不明确。庆幸的是，这只是电量低才出现的情况。

Type Markdown and LaTeX: α^2

可能的改进方向

1. 加入个人配置保存和装载（用处不是很大）
2. Feats的计算线程改良（已经改良不少，计算频率为0.08s，但我觉得应该还有空间）
3. Unregistered Feats还是用毒瘤np.append，虽然不影响计算和程序运行，但是不够丝滑，可以考虑根据appendNum的值设定Feats大小，如果Feats个数超过appendNum则清空一次，不过也不完全是这个函数引起的，还有暴力搜索等。。不过由于第二次计算切片为400，收一条数据的速度大概为0.01秒，所以整个unregistered的计算时间为4秒。。随使用。。

In []:

1

