

Google_Kaggle数据总览

1 数据处理

数据网站: <https://www.kaggle.com/c/google-smartphone-decimeter-challenge/data?select=train>

下载Data中的train数据, 其中含有groundtruth.csv数据

每个train下面都有多个手机型号数据, 为统一化和减少工作量, 尽力采用同一手机型号 (Pixel4, Pixel4XL, Pixel5居多, Mi8数据集2个)

处理后的数据: Google_Kaggle_data.zip

```
python draft.py //fix_data = glp.AndroidDerived2021("2021-03-10-US-SVL-1\\Pixel4XL\\Pixel4XL_derived.csv")对derived数据处理生成wls_estimate.csv数据
python show.py //将wls_estimate.csv和groundtruth.csv联合显示在图像上
```

show.py (来自冯海桐同学, 进行了少许修改)

tips:不同的groundtruth.csv格式可能存在差异, 需要对其进行略微修改, 找出其中的时间, 经度, 纬度, 高度数据, 导入gt_df中

```
import numpy as np
import gnss_lib_py as glp
import matplotlib.pyplot as plt
import pandas as pd
from pyproj import Proj, Transformer

# 读取 CSV 文件
wls_estimate_df = pd.read_csv("./2021-04-15-US-MTV-1/Pixel5/wls_estimate.csv")
gt_df = pd.read_csv("./2021-04-15-US-MTV-1/Pixel5/ground_truth.csv",
                    usecols=[2, 3, 4, 5], # 选择第三、第四、第五、第六列
                    names=["time", "lat_gt_deg", "lon_gt_deg", "alt_gt_deg"], #
                    # 指定列名
                    header=0) # 由于文件有表头, 设置header=0表示使用第一行作为列名

# 将 DataFrame 转换为 NavData 对象
wls_estimate = glp.NavData()
for column in wls_estimate_df.columns:
    wls_estimate[column] = wls_estimate_df[column].values








# 如果需要将转换后的数据添加到 NavData 对象中
gt_navdata = glp.NavData()
for column in gt_df.columns:
    gt_navdata[column] = gt_df[column].values

# gt1_navdata = glp.NavData()
# for column in gt1_df.columns:
#     gt1_navdata[column] = gt1_df[column].values

# gt2_navdata = glp.NavData()
# for column in gt2_df.columns:
#     gt2_navdata[column] = gt2_df[column].values
```

```
# 使用 NavData 对象进行绘图
raw_fig = glp.plot_map(wls_estimate, gt_navdata)
raw_fig.show()
raw_fig.write_html("./2021-04-15-US-MTV-1/Pixel5/compare_map.html")
```

数据处理结果显示

 supplemental	2025/1/18 19:12	文件夹
 compare_map	2025/1/20 20:11	Chrome HTML
 ground_truth	2025/1/18 19:12	XLS 工作表
 Pixel4XL_derived	2025/1/18 19:12	XLS 工作表
 Pixel4XL_GnssLog	2025/1/18 19:12	文本文档
 raw_map	2025/1/20 20:08	Chrome HTML
 wls_estimate	2025/1/20 20:08	XLS 工作表

wls_estimate: 解算得到的数据点集合

raw_map: 解算数据点的可视化结果

compare_map: groundtruth和解算数据点共存的可视化结果

Pixel4XL_derived: glp.AndroidDerived2021接口处理

Pixel4XL_GnssLog: glp.AndroidRawGnss接口处理

建议使用derived数据, GnssLog2.0无法处理, 绝大部分derived数据可以正常处理

数据量: 总计28个数据集, 其中25个为有漂移数据集, 2个为精确数据集 (Mi8数据), 1个数据集运行超时

平均数据点数: 平均每个数据集包含约1500个数据点

平均每个数据集有漂移的点可记为1000个 (有漂移数据集我感觉所有的点都在飘)

总计: $25 \times 1000 = 25000$, 按照100个漂移点左右分为1个case, 估计可分为**250个case** (可能并不均由高楼遮挡导致漂移)

2 数据分析

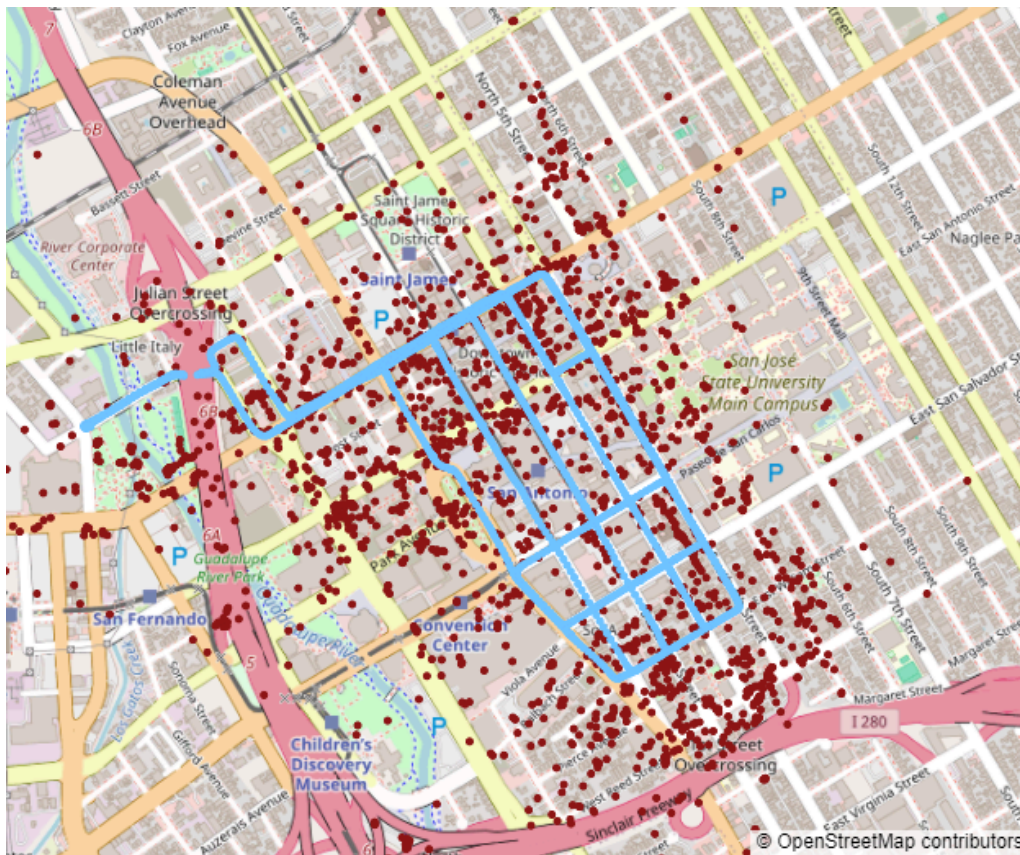
1 groundtruth

groundtruth数据由手机生成, 但是可以认为是真实数据, 都是沿着既定的路线

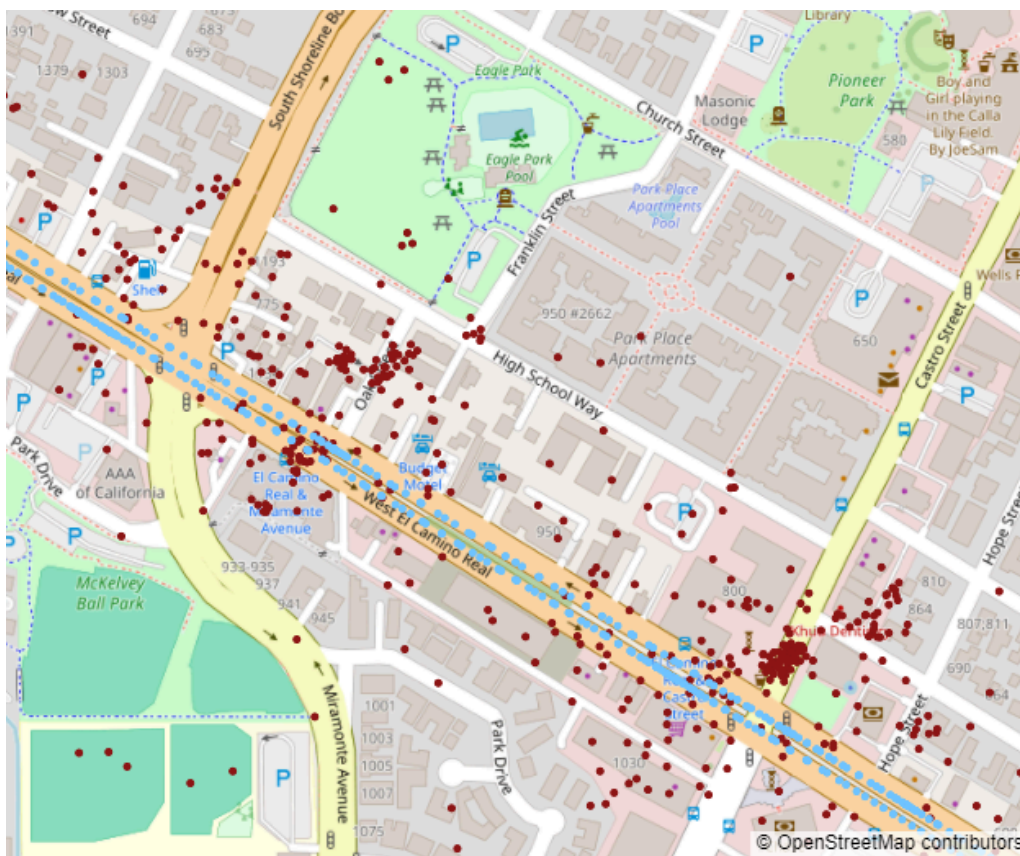
2 生成数据

Google数据的漂移现象非常普遍, 可是不好分析是否都是因为高楼导致的飘逸

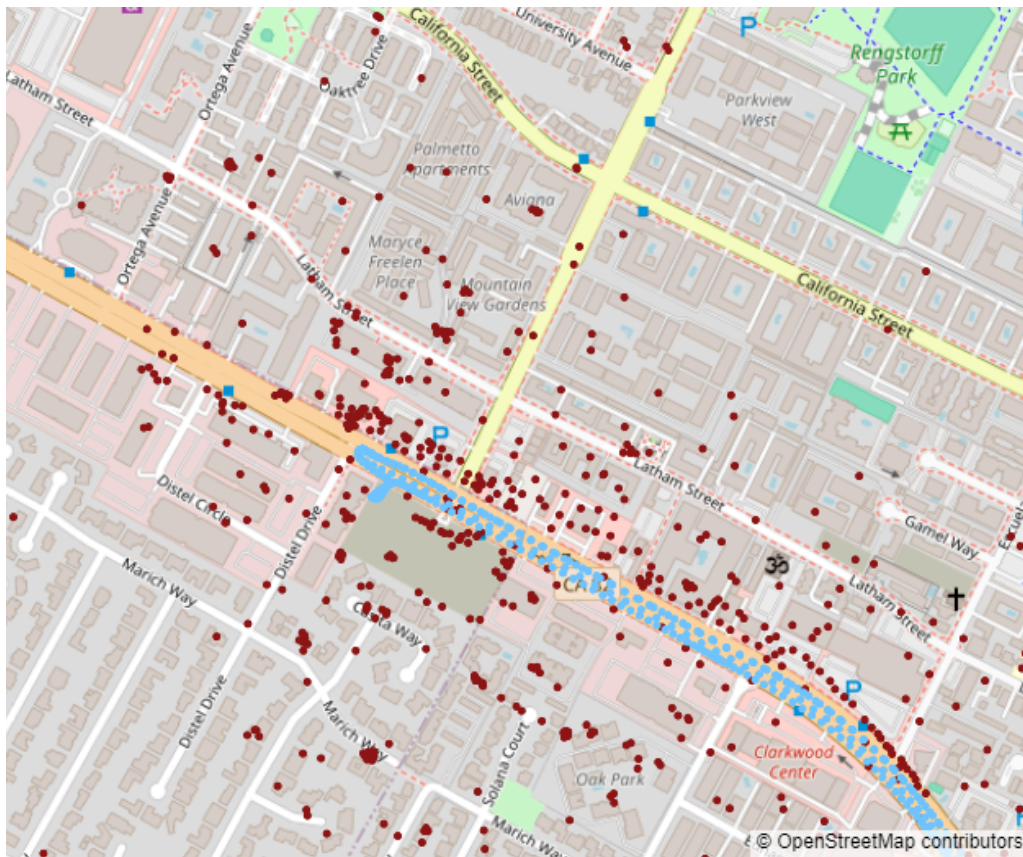
下面列举了一些漂移图例 (详细图例参考[遮挡场景图片.zip](#)) :



Trajectory
 ● RX WLS
 ● GT



Trajectory
 ● RX WLS
 ● GT



Mi8数据的拟合性非常好（可能存在价值）：

