我们分为三大部分，

**1、数据收集**

为了收集安卓手机实时位置的GPS数据，在应用方面我们使用了QPython。该应用是一个用于在Android设备上运行Python脚本的应用程序。它包含 Python 解释器、控制台、编辑器和Android脚本层(Scripting Layer for Android, SL4A)，并且支持许多常见的Python库和模块，使用户能够在移动设备上编写和运行Python代码。QPython分为QPython2和QPython3，他们分别基于Python2和Python3，我们使用的是Python3。

在收集数据时我们用的是androidhelper库，androidhelper库是对SL4A的封装。SL4A与Android脚本环境(Android Scripting Environment, ASE)意义相同，据Google官方介绍，SL4A将脚本语言带入Android，允许用户编辑和执行脚本，是一个可以直接在Android设备上运行交互式解释器。

首先创建一个实例，调用startLocating方法开始收集定位数据，并设置更新之间的最短时间minDistance（以毫秒为单位）为1000，也就是1秒。因为在收集数据时我们的手机设备是静止不动的，为了收集有效数据，我们设置更新之间的最短距离minUpdateDistance（以米为单位）为0。然后调用readLocation方法获取定位数据，该方法返回的是一个字典，这个字典包含了来自三个不同提供者（'network'、'gps'、'passive'）的位置信息，这里我们需要的是GPS的数据。但是这个字典的具体的结构可能会有所变化，这它取决于设备的硬件和设置，也就是说不是每次调用这个方法都能获得GPS提供的数据。因此我们用了try-except语句来捕获异常，当返回的字典不包含GPS数据时进行跳过，直至能够获取GPS数据，与此同时我们设置了5秒的休眠时间，也就是每隔5秒收集一次数据。而在GPS数据中包括的字段有：altitude海拔高度、latitude纬度、longitude经度、time位置信息的时间戳、accuracy位置信息的精确度、speed移动速度、bearing移动方向。其中，accuracy 表示位置信息的精确度，即该位置与实际位置之间的误差，单位通常是米。通常情况下，较低的 accuracy 值表示位置信息更为准确；speed 表示设备的移动速度，通常以米每秒（m/s）为单位。这个值表示设备在获取位置信息时的瞬时速度。bearing 表示设备的方向角，也称为方向角度，表示设备当前位置相对于某个方向的角度。在地理导航中，这个方向通常与地理北方向（True North）之间的夹角。0度通常表示北方，90度表示东方，180度表示南方，270度表示西方。bearing 的具体计算方式可能取决于设备的定位提供者和设备的运动状态。当设备是通过 GPS 获取位置信息时，bearing 可能是相对于真北方向的角度。在实际应用中，bearing 的信息对于导航和定位是非常有用的，因为它可以指示用户当前位置朝向的方向（字段介绍部分可能还得看具体用到什么才决定某些东西要不要写那么详细）。这里我们用到的数据有海拔高度、纬度、经度和精准度。

**2、数据处理**

在GPS定位中，涉及到距离、速度、时间等不同的物理量，它们可能具有不同的量纲（单位）。为了进行有效的计算和数据处理，需要对这些量纲进行适当的处理和统一。

而且，后面我们会涉及到多边定位，所以需要统一量纲，我们需要使用卫星与接收器之间的距离数据。为了确保这些距离数据具有一致的量纲，可以选择参考点，将所有距离数据与该参考点之间的距离进行归一化。这样可以简化计算，并减小浮点数运算中的舍入误差。

在进行坐标的最大最小化时，通常是指将一组坐标值缩放到特定的范围或区间内。这种处理常用于数据标准化、特征缩放等应用场景，可以确保不同坐标值之间具有可比性，并且有助于提高模型的性能。

在进行数据处理时，对重复的行数据进行去重处理，保留有效并且不重复的数据进行使用。其次对转换后的ECEF三维坐标进行标准化，我们采用的方法是最大最小值标准化，将坐标值压缩在[0,1]的区间内，公式如下：

其实就是遍历所有的坐标值，找到各自的最大值和最小值。最大值表示原始坐标中的最大值，最小值表示原始坐标中的最小值。这样是为了可以提高模型的解释性，确保模型在求解时不会出现奇异矩阵的情况。例如，当坐标的取值范围非常大时，模型的系数或权重可能变得非常小，难以直观解释。通过将坐标进行最大最小化，可以使得模型的系数或权重具有更好的可解释性。

**3、模型建立和求解**

根据收集到的数据坐标，建立卫星导航系统的数学模型。

我们在手机端收集到的数据中发现，数据里显示的是经度（Longitude）、纬度（Latitude）和高度（Altitude）的数值，这是人们所用到的测地坐标系的坐标，在接收机定位解算的过程中，卫星的位置以及用户的位置都是在 ECEF 坐标系内表示的，但是，人们一般习惯使用经度和纬度来表示位置，因此，需要实现 ECEF坐标系和测地坐标系之间的转换。那么，根据文献可知，我们可以通过以下的关系从测地坐标系到 ECEF 的转换：

其中，表示纬度，表示经度，表示海拔高度。

转换完之后，我们需要进行量纲的处理，还有对坐标进行标准化。刚刚的数据处理中所提到。

接着，我们根据这样的一个多边定位示意图进行求解。而为什么我们会选择多边定位求解，首先，多边定位是一种基于测量多个点与目标之间距离或方向的方法，通过这些测量结果来确定目标的位置。它是一种常用的定位技术，广泛应用于地理测量、导航系统（如GPS）以及其他需要确定物体位置的领域。

而它的原理是利用几何三角形的性质，通过测量多个已知位置的点与目标点之间的距离或方向，来计算目标点的位置。在二维空间中，至少需要三个已知点与目标点的距离或方向信息；在三维空间中，至少需要四个已知点与目标点的距离或方向信息。

因为它的优势是可以在没有先验信息的情况下确定目标位置，只需要测量点与目标之间的距离或方向即可。所以我们用其来求解。

而为了求得误差项更小，我们用了最小二乘法进行求解。它的求解思路如下：

1. **带有未知参数的表示方程组：**

假设待定点为，该点是无限接近我们真实所处的位置。根据点与点之间的距离公式，可得的表示方程组如下：

其中，表示待预测点与检测点的距离。这里是使用了数据中accuracy字段作为。

根据上述方程组，将第一个方程减去第二个方程，第二个方程减去第三个方程，以此类推，达到消除方程组中位置参数的平方次幂：

此方程组的矩阵表示形式为：，即：

1. **定义误差：**

由于在个方程中，待求点坐标不会符合方程组中的所有方程，因此设置误差向量为：。取误差向量中误差的平方和，则有：

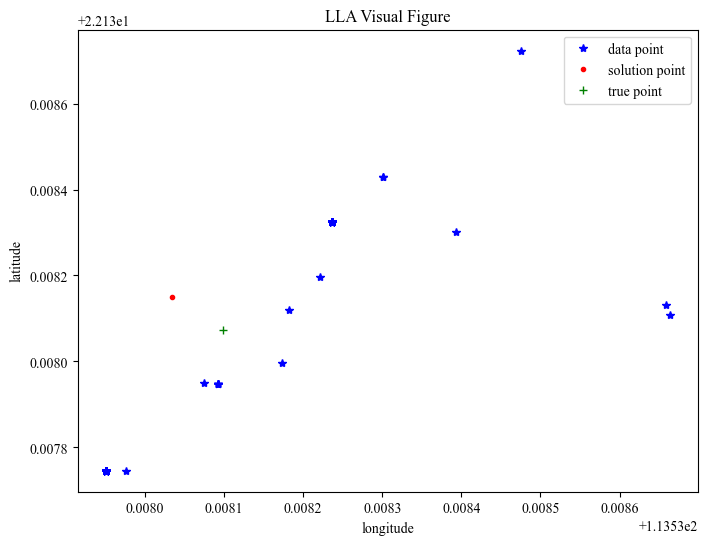
1. **使用最小二乘法求解**

根据最小二乘法定义，若使误差最小，需要根据上式对求导，令其导数为0，得到表达式如下：

求解得到：

因此可以获得待定点的坐标：。后续根据数据预处理的信息，将求解结果进行逆标准化，还原得到真实的ECEF坐标，最终转换成纬度、经度、海拔高度。

最终求得纬度、经度、海拔高度为：。将其与观测点经纬度绘制成二维平面图如下：



**4、遇到的困难，改进与不足**

**一、**起初收集demo数据时，经过去重得到的可用数据为4个观测点，其海拔高度都是相同的，初期剔除了海拔仅使用经纬度进行实验，仅仅将经纬度转换成二维的坐标，最后得到的结果在平面理想情况下是接近真实地点的。因此后续重新收集了数据，得到多个海拔高度、经度、纬度不一并且接近真实地点的数据，查阅资料使用了ECEF的椭球模型右手三维坐标系，将经纬度和海拔转换到了三维坐标上，从而帮助我们完成了实验。

二、经过经纬度转换成三维坐标后，使用最小二乘法求解的结果并不是特别理想，预测的坐标甚至比观测点的距离还要长，因此在这里引入了Min-Max标准化，将转换的三维坐标和距离进行标准化，最终求出来的待定点坐标更接近真实坐标了。

三、在本次实验中，我们收集的数据缺乏来自GPS卫星的信息，因此无法求得伪距、时钟偏差等更详细的数据，因此在最后最小二乘求解上缺乏更多可靠的数据来源。

四、收集的数据中，对于accuracy这一列数据的真实性有待考究，其描述的究竟是观测点和真实点的直线距离还是地表距离，未经过转换直接使用是否会造成最终结果误差。

一、确定目标函数：首先，需要确定要最小化的目标函数。在最小二乘法中，目标函数通常是衡量模型预测值与实际观测值之间差异的平方和。而在这里，我们利用未知参数表示的观测值的方程。

二、建立数学模型：根据具体问题，建立一个数学模型来描述自变量和因变量之间的关系。这个模型可以是线性的或非线性的，具体形式取决于问题本身。而我们这个问题是线性的。

三、定义残差：在最小二乘法中，残差是指模型预测值与实际观测值之间的差异。它是目标函数的关键部分。现在误差方程为非线性（二次方程），必须转成一次方程得到简化形式才可以使用最小二乘法原理。我们令、分别为待定坐标的近似值和修正值。我们根据一元函数在点处的泰勒展开式，取其一次项（降幂），得到一次方程，然后联立方程组，化简后可以得到误差方程：V=AX+L。