

**Lab1实验报告**

|  |  |
| --- | --- |
| **题目：** | **无线信号感知与室内定位** |

|  |  |
| --- | --- |
| **小组成员：** | 陈喆桓 1900013017 |
|  | 郭一凡 1900011755 |
|  | 许轩卓 1800013038 |
|  | 王歆宇 1800017869 |

**摘要**本实验完成了基于Wi-Fi的室内三角定位方法。实验通过3个Wi-Fi嗅探器采集待测设备的Wi-Fi信号，搭建了服务器与数据库MySQL用于对采集到的Wi-Fi信号进行接收和存储。算法使用由Wi-Fi嗅探器采集的RSSI值得出的物理距离和嗅探器的位置坐标计算出待测设备的坐标。定位的准确度较为一般，分析了导致定位准确度较低的可能因素，认为RSSI值测量的准确程度将显著影响定位精度与准确度。实验代码已在<https://github.com/ACMLCZH/Network_Lab/tree/main/Lab1> 上开源

**关键词** 三角定位法；Wi-Fi感知；RSSI信号

[1 服务器的搭建 3](#_Toc102857427)

[1.1 服务器架构 3](#_Toc102857428)

[1.1.1 TCP连接 3](#_Toc102857429)

[1.1.2 多线程并发 3](#_Toc102857430)

[1.2 报文接收 3](#_Toc102857431)

[1.2.1 数据提取 3](#_Toc102857432)

[1.2.2 长报文处理 3](#_Toc102857433)

[2 数据库的搭建 3](#_Toc102857434)

[2.1 基本配置信息 3](#_Toc102857435)

[2.2 数据库搭建流程 4](#_Toc102857436)

[3 基于Wi-Fi信号的定位计算 4](#_Toc102857437)

[3.1 定位计算目标 4](#_Toc102857438)

[3.2 定位计算原理 4](#_Toc102857439)

[3.2.1 三角定位算法的基本原理 4](#_Toc102857440)

[3.2.2 通过RSSI测距的基本原理 5](#_Toc102857441)

[3.3 算法部分的代码说明 5](#_Toc102857442)

[3.4 数据采集 6](#_Toc102857443)

[3.4.1 工作曲线参数Pr(d0)与n的标定 6](#_Toc102857444)

[3.4.2 三角定位法的数据采集 7](#_Toc102857445)

[3.5 数据预处理 7](#_Toc102857446)

[3.6 数据与结果 8](#_Toc102857447)

[3.6.1 待测设备静止时的数据处理 8](#_Toc102857448)

[3.6.2 待测设备移动时的数据处理 8](#_Toc102857449)

[3.7 分析与讨论 8](#_Toc102857450)

[3.7.1 待测设备静止时定位结果的准确性 8](#_Toc102857451)

[3.7.2 待测设备移动时定位结果的展示 11](#_Toc102857452)

[4 实验结论 12](#_Toc102857453)

1 服务器的搭建

本次实验中需要用到服务器来接收3个Wi-Fi嗅探器捕捉到的信息。该部分由王歆宇、陈喆桓同学完成。

1.1 搭建工具

搭建服务器及数据处理所使用的python脚本库主要有：

socket：创建TCP套接字，让服务器完成监听、建立连接、接收数据等操作；

threading：使用多线程并发接收数据，并使用线程锁保护MySQL的读写安全；

json：用于解析数据字符串，提取数据。

1.2 服务器架构

1.2.1 TCP连接

在本次实验中，服务器需要接收来自嗅探器的HTTP报文，因此需要利用TCP套接字建立TCP连接，绑定至特定的端口，之后在端口上进行监听。

server = socket(AF\_INET, SOCK\_STREAM)

server.bind((Server\_IP, Server\_port))

server.listen(N)

其中AF\_INET参数表示底层网络使用的是IPv4，SOCK\_STREAM表示这是一个TCP连接，N表示最大可排队的连接数量。

1.2.2 多线程并发

在实验中，同时有多台嗅探器向服务器发送报文，因此采用多线程并发方式进行处理。主线程持续等待来自客户端的连接请求，当主线程接收到连接请求时，其与发起连接请求的嗅探器（客户端）的端口进行三次握手建立TCP连接。然后主线程将TCP连接交给子线程，待子线程启动后，主线程返回继续等待另一个客户端的连接。

**1.3 报文接收和数据提取**

1.3.1 提取实体

接收到的HTTP报文以“POST”开始，数据包含在报文最后的实体（content）部分，与报文首部以空行（”\r\n”）隔开。将实体提取并交付给Wi-Fi数据提取程序。

1.3.2 长报文处理

在数据报接收过程中，需要注意客户端在发送HTTP数据报时，对于长度超过1024字节的数据报，首先只发送数据报首部，并在首部中加入“Expect:”字段，内容为“100-continue”，表示数据报长度较长，需要等待服务器的许可，才能发送数据正文部分。因此对于搭建的服务器，需要在接收数据时检查接收数据报的首部中是否附有“Expect:”字段，如果附有，则应回复“100 Continue”以通知客户端继续发送数据正文。

1.3.3 提取Wi-Fi数据

实体内容为”data=”+json格式的字符串，因此只需使用json工具包将字符串反序列化即可。在其中提取出需要的字段（见2.2部分），过滤掉噪声数据（如非追踪设备MAC地址传来的信号）后交付给数据库。注意到MySQL的数据读写是非线程安全的，所以还需要采用线程锁来保证读写安全。

2 数据库的搭建

本实验中需要用到数据库来记录Wi-Fi嗅探器发送服务器的信息。该部分由许轩卓同学完成。

2.1 基本配置信息

数据库使用了MySQL-8.0.28，运行环境为Windows11。其中python脚本部分主要使用的是pymysql包，该库提供了封装MySQL数据库操作的供python程序调用的接口函数。

2.2 数据库搭建流程

首先建立数据库。具体的建立指令不加赘述。数据库中只包含所需要的字段，分别为：wifi\_id（嗅探器的id）, wifi\_mac（嗅探器的MAC地址）, collect\_time（收集到此条数据的时间）, device\_mac（追踪设备的MAC地址）, device\_rssi1（追踪设备的RSSI值）。对于数据进行必要的预处理之后，我们将所需要的数据按照键的顺序组成一个元组。利用python的格式化字符串可以将我们需要的值方便地构建成MySQL语句。执行构建好的MySQL语句，便可以将嗅探器获得的数据插入到数据库中。

3 基于Wi-Fi信号的定位计算

这一部分阐述了定位算法的原理与具体实现。为了得到定位计算的数据，首先需要进行实地的数据采集，该部分由小组成员共同完成。

完成数据采集后，需要进行预处理以得到可供算法直接使用的数据。最后展示并分析了定位的结果。以上部分以及算法的设计由郭一凡同学完成。

3.1 定位计算目标

通过算法实现对确定MAC地址设备的位置计算。算法能够根据3个Wi-Fi嗅探器的坐标以及它们与待测设备之间的距离计算出待测设备的坐标，从而实现对该设备的定位。

3.2 定位计算原理

3.2.1 三角定位算法的基本原理

如**图1**所示，设3个信号发射端A、B和C的坐标分别为(xa, ya)，(xb, yb)和(xc, yc)，设目标点D的坐标为(x, y)。通过RSSI信号强度与物理距离的转换，可知A、B和C与D之间的距离分别为dA，dB和dC。有如下方程组(\*)：



**图1 三角定位算法的原理**

首先对(\*)中的前两式分别平方后相减，得：

同理有：

两方程组相减并移项后得到方程组(\*\*)：

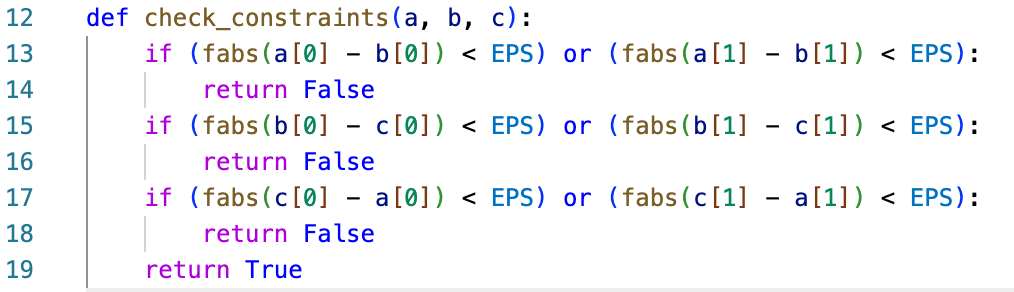
算法的主要部分是对(\*\*)方程组中*x*, *y*的计算。

3.2.2 通过RSSI测距的基本原理

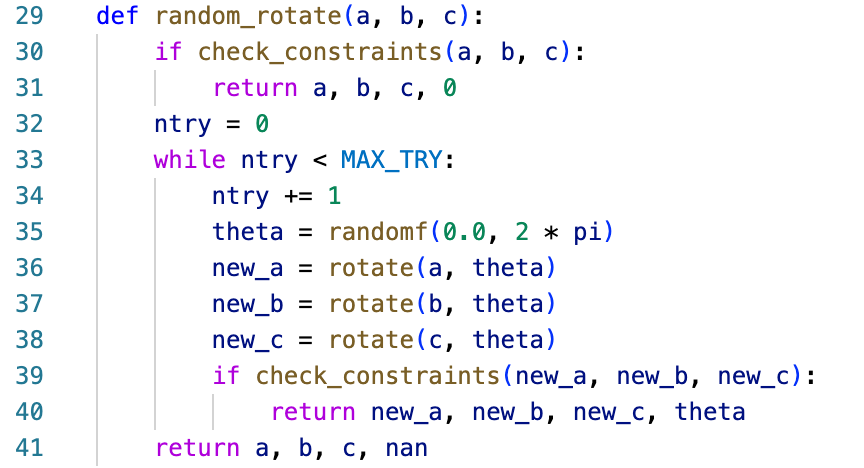
由于Wi-Fi嗅探器自身提供的物理距离有很高的不准确性，因此在实验中只采用了Wi-Fi嗅探器提供的RSSI值，通过RSSI信号强度与物理距离的转换公式计算出实际距离，最终以计算出的距离用于位置坐标的计算。RSSI值与距离之间有如下关系：

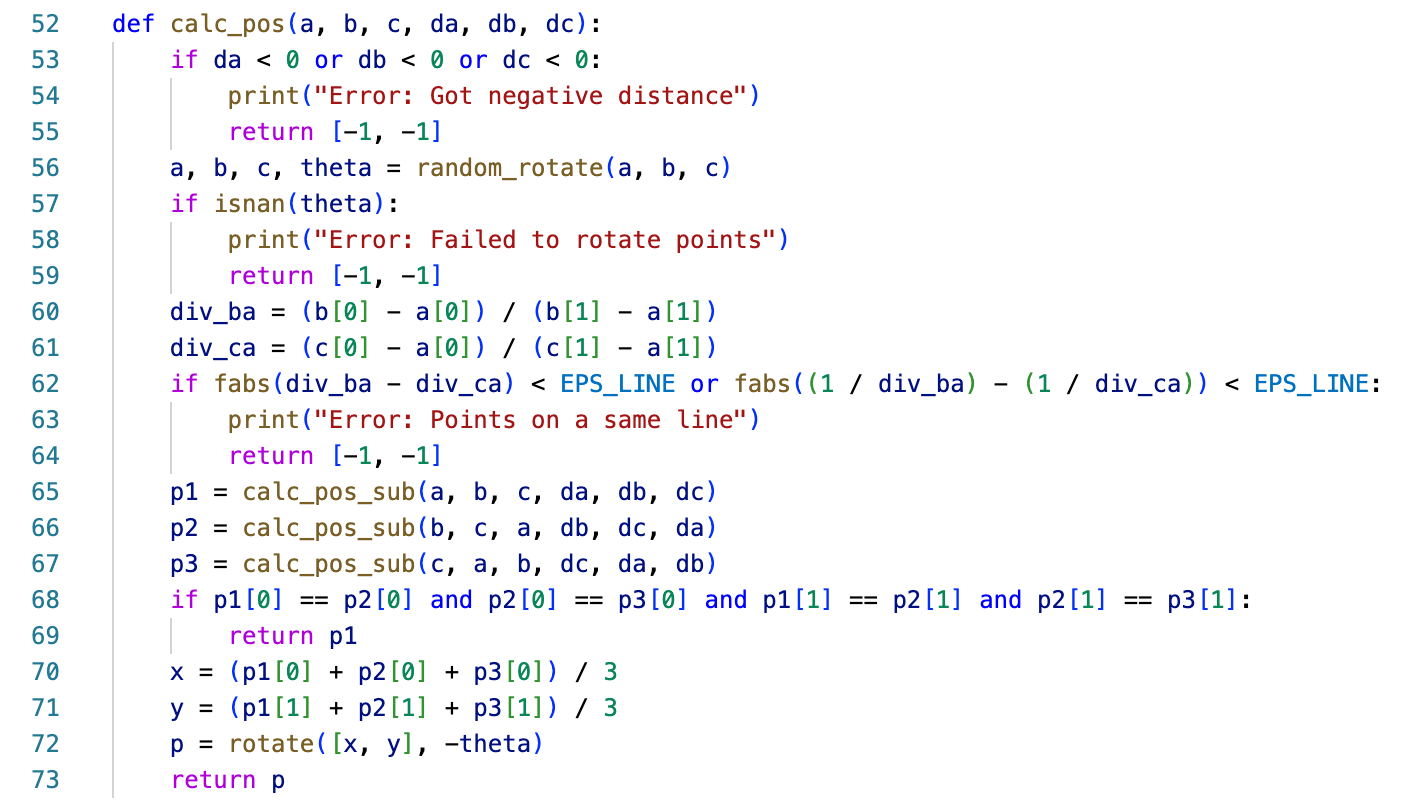
其中，为收发端相距*x* m时的信号强度(dB)；n为环境衰减因子，无量纲；d为收发端的物理距离，单位为m。由于公式中的参数n与Pr(d0)会受到物理环境的影响，因此需要进行标定。

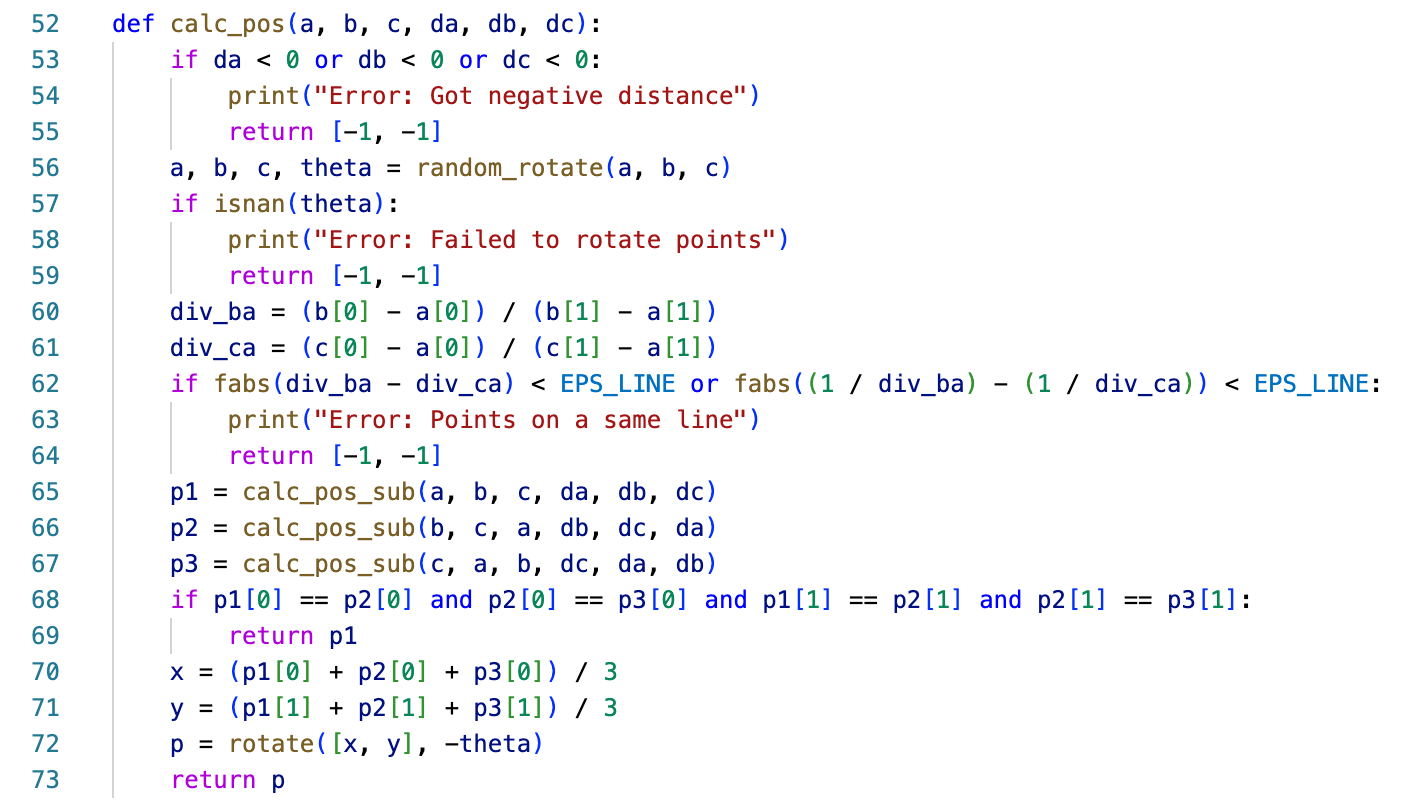
3.3 算法部分的代码说明

 根据方程组(\*\*)可知，*x*, *y*表达式中的分母不能为0，也即不能出现以3个Wi-Fi嗅探器为顶点的三角形的某一条边与选定的坐标系的坐标轴平行的情况。代码中首先对这种情况进行了判断：

当出现边平行与坐标轴的情况时，需要对输入的点进行随机旋转，以使方程组(\*\*)能够被计算：



 接下来根据输入的坐标点与距离进行待测设备坐标的计算。首先分别判断了输入距离为负值、旋转失败以及3个Wi-Fi设备共线的特殊情况，而后根据方程组(\*\*)计算出了待测设备的坐标，返回将各个点旋转回初始位置后的坐标值。



3.4 数据采集

**图2 采集场景**

采集工作地点为农园食堂二层（如**图2**）。采集内容分为两组，一组用于对工作曲线参数的标定，另一组用于三角定位算法的测试，采集的具体过程如下：

3.4.1 场景布置

（1）选取无人时段来到采集地点，并且尽量清理桌面上的隔板，减少信号干扰；

（2）根据采集地点的情况，选取坐标原点，建立平面直角坐标系；

（3）确定3个Wi-Fi嗅探器的摆放位置（如**图2**中红圈位置）。在实际测量中，我们选取瓷砖整点的正上方（每块瓷砖规格为60cm×60cm的正方形）作为嗅探器摆放位置，并且将其均放置于桌面上，保证嗅探器位于同一水平面。

3.4.2 工作曲线参数Pr(d0)与n的标定

（1）记录3个Wi-Fi嗅探器的相对位置（坐标）；

（2）在以某个设备为圆心、半径为0.2、0.5、1、1.5、2、3 m的圆周上测量多个RSSI值，记录下设备名称与相应的RSSI值。

3.4.3 三角定位法的数据采集

1. 待测设备静止时的数据采集

（1）保持3个Wi-Fi嗅探器的位置不变，确定待测设备的相对位置，并记录；

（2）保持接收设备静止，采集一段时间的数据并记录；

（3）改变接收设备的位置，重复上述步骤，记录数据。

2. 待测设备移动时的数据采集

保持3个Wi-Fi嗅探器的位置不变，手持待测设备使之按照规划好的路线匀速缓慢移动，过程中采集数据并记录。

3.5 数据预处理

以设备45为圆心，分别在不同半径的圆周上采集数据约1 min，对采集到的RSSI值求平均，得到相应距离下的RSSI信号强度，数据如**表1**所示。

**表1 不同距离对应的RSSI信号强度**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 距离d/m | 0.2 | 0.5 | 1 | 1.5 | 2 | 3 |
| RSSI信号强度Pr(d)/dB | -30.05 | -34.61 | -45.00 | -50.43 | -55.30 | -58.25 |

根据**表1**中的数据，绘制Pr(d)-log d 图，可得到回归方程：

根据截距和斜率可分别求出Pr(d0)与n。拟合出的工作曲线如**图3**所示。



**图3 工作曲线的拟合**

根据回归方程可知，，。

3.6 数据与结果

3.6.1 待测设备静止时的数据处理

选取好坐标系后，将3个Wi-Fi嗅探器分别固定在坐标为(2.4, 1.2)，(6.0, 5.4)和(0, 10.8)的位置（单位：m），分别将待测设备固定在6个不同的位置采集数据约1 min。对于每个设备，将其在每个位置采集到的RSSI值分别取平均，根据工作曲线将RSSI值转换为实际距离。每个位置下的实际位置坐标、通过RSSI值计算出的Wi-Fi嗅探器到待测设备的距离以及通过算法计算出的位置坐标如**表2**所示。

**表2 静态数据采集与处理（单位：m）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 待测设备坐标 | 设备31测得的距离 | 设备45测得的距离 | 设备67测得的距离 | 计算出的坐标 |
| 1 | (2.4, 5.4) | 3.011 | 2.690 | 7.929 | (3.812, 3.851) |
| 2 | (4.8, 9.6) | 7.541 | 2.163 | 3.861 | (4.722, 9.066) |
| 3 | (4.8, 4.8) | 3.669 | 4.152 | 5.953 | (1.712, 4.983) |
| 4 | (4.8, 2.4) | 3.594 | 2.333 | 9.479 | (5.507, 3.070) |
| 5 | (1.8, 2.4) | 2.069 | 4.060 | 7.789 | (2.424, 3.369) |
| 6 | (1.8, 4.8) | 3.130 | 5.228 | 5.732 | (0.283, 4.570) |

3.6.2 待测设备移动时的数据处理

保持3个Wi-Fi嗅探器的位置不变，手持待测设备，从点(1.8, 9.6)开始向y轴负方向移动至点(1.8, 2.4)，而后向x轴正方向移动至点(5.4, 2.4)结束（单位：m），全程共约3 min。首先对采集到的数据进行关于距离的线性插值以得到每秒的位置信息，而后使用滑动窗口的方法进行平滑，设置窗口大小为5个数据点（即5 s），使用同样的算法对插值并平滑后的数据进行处理，最终得到计算出的轨迹。

3.7 分析与讨论

3.7.1 待测设备静止时定位结果的准确性

为了更方便地讨论定位结果的准确度，设置“绝对偏差”和“相对偏差”两个指标。其中绝对偏差指待测设备的实际位置与实验计算出的位置之间的物理距离，即：

相对偏差指绝对偏差与待测设备和与之相距最远的Wi-Fi嗅探器之间的距离的比值，即：

根据**表2**中的数据，可分别计算出每次测量的绝对偏差与相对偏差，计算结果如**表3**所示。

**表3 静态数据的绝对偏差与相对偏差**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 待测设备坐标/m | 计算出的坐标/m | 绝对偏差/m | 相对偏差 |
| 1 | (2.4, 5.4) | (3.812, 3.851) | 2.10 | 0.355 |
| 2 | (4.8, 9.6) | (4.722, 9.066) | 0.54 | 0.062 |
| 3 | (4.8, 4.8) | (1.712, 4.983) | 3.09 | 0.403 |
| 4 | (4.8, 2.4) | (5.507, 3.070) | 0.97 | 0.101 |
| 5 | (1.8, 2.4) | (2.424, 3.369) | 1.15 | 0.134 |
| 6 | (1.8, 4.8) | (0.283, 4.570) | 1.53 | 0.245 |

由**表3**可见，位置计算的绝对误差普遍为1~2 m，误差较大时可达约3 m，误差较小时为约0.5 m；相对误差普遍为10%~20%，误差较大时可达约40%，误差较小时约6%。整体而言，定位结果的准确度较为一般。**图4**中给出了更为直观的定位结果，其中同色点为一组，每组中“·”代表实际位置，“x”代表计算出的位置。



**图4 静态数据定位的可视化结果**

为了分析定位结果准确度不足的原因，在**图5**中直观地给出了每个待测位置下的定位计算过程。其中红色点代表Wi-Fi嗅探器点位置；蓝色大圆（弧）以Wi-Fi嗅探器为圆心，以通过RSSI值计算出的距离为半径；黑色点代表待测设备的实际位置；橙色点为定位算法计算出的位置。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (a) (2.4, 5.4) | (b) (4.8, 9,6) | (c) (4.8, 4.8) |
|  |  |  |
| (d) (4.8, 2.4) | (e) (1.8, 2.4) | (f) (1.8, 4.8) |

**图5 静态数据定位计算过程的可视化**

由**图5**可见，在多数情况下，3个蓝色的大圆能够趋近于交于一点，如(a)，(c)，(d)，(e)，(f)；少数情况下，3个大圆的分布较分散，不能形成有效的相交，如(b)。根据前文中对于算法原理的分析可知，若3个圆的排列越趋近于恰好相交于一点，则算法计算出的位置越接近真实位置。因此，仅从算法的角度看，该算法在各种情况下都能够较为准确地根据输入的嗅探器坐标和收发端距离预测待测设备的位置坐标。

接下来考察收发端之间的距离测量对定位结果的影响。首先考虑RSSI信号强度到距离的转换公式中的参数标定引入的误差。为简化讨论，暂忽略实验过程中Wi-Fi嗅探器测量RSSI值的误差。静态数据测量的RSSI值在-71~-54 dB之间分布，则根据**图3**中的拟合曲线参数误差可计算出，由拟合曲线参数误差导致的距离测量绝对误差在± 0.3~± 3 m之间分布，相对误差在14%~32%之间分布，且测量的距离越长，误差值越高。此外，实验中发现，当保持收发端距离相同时，不同时间下测得的RSSI值有一定区别，故Wi-Fi嗅探器对RSSI值的测量也有一定的不准确性。因此，RSSI信号的采集以及向物理距离的转化都有较高的不准确度，这将使定位结果的准确性显著下降。

推测实验中可能导致RSSI信号及转化出的物理距离准确度不高的原因有：

(1) 工作曲线所覆盖的距离范围较小。在工作曲线的测定中，仅覆盖到了0.2~3 m的距离，而实验中涉及的最长距离约为8 m。若扩大工作曲线的覆盖范围，可能能够更加真实地反映出本次实验条件下RSSI值与物理距离的关系，从而减小距离测量导致的误差；

(2) 测试环境中存在较多物理干扰。尽管实验中已尽量移开各种物理障碍，但由于测试地点农园二层有很多的桌椅与就餐隔板，可能会反射、衰减信号；导致环境中存在较多干扰信号，从而影响Wi-Fi嗅探器对真实信号的捕捉。此外，其他带有Wi-Fi功能设备的干扰使得嗅探器采集到的实验设备信息不完整且在时间上不连续，也将影响嗅探器对真实信号的捕捉。

3.7.2 待测设备移动时定位结果的展示

手持设备移动的定位结果如**图6**所示。其中红色点代表3个Wi-Fi嗅探器的位置；橙色轨迹为待测设备实际的移动轨迹；蓝色轨迹为定位算法计算出的移动轨迹。动态效果点击链接：<http://162.105.162.212:7506/html/result.html> （前端实现由陈喆桓同学完成）

从**图6**中可大致看出，计算出的轨迹覆盖区域与实际的轨迹覆盖区域有大部分重合，但很难反映待测设备真实的移动情况。



**图6 动态数据定位的可视化结果**

4 实验结论

本实验使用Wi-Fi三角定位算法对给定MAC地址的设备进行定位，并分别测试了算法对于静态数据和动态数据的定位准确性。在两种数据下，定位结果的准确性一般。其中RSSI值的测量不准确度和RSSI到物理距离转换的不准确度对定位误差的贡献较大，算法的不准确度对定位误差的贡献很小。