**大连理工大学本科毕业设计（文）**

**基于蓝牙的室内定位系统设计与实现**

**The research and Implementation of Indoor Location Positioning System Based on Bluetooth**

学 院（系）： 软件学院

专 业： 软件工程

学 生 姓 名： 张冬润

学 号： 201392221

指 导 教 师： 赖晓晨

评 阅 教 师：

完 成 日 期：

大连理工大学

Dalian University of Technology

# 摘 要

长久以来，人们对定位系统的研究从未停歇，无论室内还是室外，如何快速准确并且低成本的进行定位成为了主要的研究方向。针对室内定位需求增加，定位方法多种多样且无普适方法的情况设计了基于蓝牙搭建的一套室内定位系统。

蓝牙作为短距离数据传输规范，在全球范围内都得到了普遍支持。另一方面，随着蓝牙4.0的发布，低功耗、低成本的特性使得蓝牙定位系统更易推广使用，可以有效的利用现有的硬件投入。

模糊指纹算法是室内定位算法的一种，室内环境下蓝牙信号传播的多径效应非常明显，模糊指纹算法可以最大程度的减少周围环境对定位结果的影响，使得定位结果更加准确。

该系统具有节能高效、环境适应能力强等特点。依托更加高效的蓝牙4.0模块，感应距离大，单个电池使用时间更长；针对室内复杂的环境，采用模糊指纹算法：先采集数据建立离线数据库，定位时采集数据进行比对，直接得出位置信息。最大化的排除复杂的环境带来的影响。手机端、服务器的C/S模式能够使得系统使用更加方便。

关词**：室内定位；蓝牙4.0；位置指纹； RSSI**

**indoor position System Design Based On Bluetooth**

# **Abstract**

For a long time, the research on the positioning system has never stopped. Whether indoor or outdoor, how to locate quickly, accurately and at low cost has become the main research direction. A set of indoor positioning system based on Bluetooth is designed according to the increasing demand for indoor location, the variety of localization methods and the absence of universal methods.

As a short distance data transmission specification, Bluetooth has been widely supported all over the world. On the other hand, with the release of Bluetooth 4.0, low power consumption and low cost make Bluetooth positioning system easier to use, and can effectively make use of the existing hardware input.

The fuzzy fingerprint algorithm is a kind of indoor positioning algorithm, indoor multipath effect of Bluetooth signal propagation is very obvious, the fuzzy fingerprint algorithm can reduce the maximum impact of the environment on the positioning results, so that more accurate positioning results.

The system has the characteristics of energy saving, high efficiency and strong adaptability to the environment. Relying on more efficient Bluetooth 4 module, sensing distance, a single battery for longer periods of time; for complex indoor environment, using the fuzzy fingerprint algorithm: first, collecting data to establish an offline database, positioning data alignment, position information obtained directly. Maximize the impact of removing complex environments. The C/S mode of the mobile terminal and server can make the system more convenient.

**Key Words：Write Criterion；Typeset Format；Graduation Project (Thesis)**

目 录

[摘 要 I](#_Toc483243731)

[Abstract II](#_Toc483243732)

[引 言 1](#_Toc483243733)

[1 绪论 2](#_Toc483243734)

[1.1 研究背景及研究意义 2](#_Toc483243735)

[1.3 研究现状及发展趋势 2](#_Toc483243736)

[1.3 本文研究内容 4](#_Toc483243737)

[1.4 章节安排 4](#_Toc483243738)

[2 蓝牙室内定位相关技术 4](#_Toc483243739)

[2.1 常见室内定位算法 4](#_Toc483243740)

[2.1.1 非测距算法 4](#_Toc483243741)

[2.1.2 基于测距的算法 5](#_Toc483243742)

[2.1.3 节点定位算法 5](#_Toc483243743)

[2.3 蓝牙概述 7](#_Toc483243744)

[2.2.1 蓝牙技术的特点 7](#_Toc483243745)

[2.2.2 蓝牙技术的应用 8](#_Toc483243746)

[2.2.3 蓝牙协议体系结构 8](#_Toc483243747)

[2.4 spring框架](#_Toc483243748)[概述 12](#_Toc483243748)

[2.5 Matlab概述 13](#_Toc483243749)

[3 需求分析 13](#_Toc483243750)

[3.1 假定和约束 14](#_Toc483243751)

[3.2 功能需求 14](#_Toc483243752)

[3.2.1 软件需求 14](#_Toc483243753)

[3.2.2 硬件需求 15](#_Toc483243754)

[3.3 未来可扩展需求 15](#_Toc483243755)

[4 定位系统设计 16](#_Toc483243756)

[4.1 硬件设计 16](#_Toc483243757)

[4.2 算法设计 16](#_Toc483243758)

[4.2.1 位置指纹 16](#_Toc483243759)

[4.2.2 加权K近邻算法 16](#_Toc483243760)

[4.2.3 模糊C均值聚类划分 17](#_Toc483243761)

[4.3 软件设计 18](#_Toc483243762)

[4.3.1 服务器端设计 18](#_Toc483243763)

[4.3.2 移动端设计 20](#_Toc483243764)

[5 定位系统实现 20](#_Toc483243765)

[5.1 硬件实现 20](#_Toc483243766)

[5.2 算法实现 21](#_Toc483243767)

[5.2.1 位置指纹 21](#_Toc483243768)

[5.2.2 加权K近邻算法 21](#_Toc483243769)

[5.2.3 模糊C均值聚类划分 24](#_Toc483243770)

[5.3 软件实现 25](#_Toc483243771)

[5.3.1 移动端实现 25](#_Toc483243772)

[5.3.2 服务器端实现 28](#_Toc483243773)

[6 定位系统测试 30](#_Toc483243774)

[6.1 测试理论 30](#_Toc483243775)

[6.2 测试结果 30](#_Toc483243776)

[6.2.1 硬件测试 30](#_Toc483243777)

[6.2.2 软件](#_Toc483243778)[测试 32](#_Toc483243778)

[结 论 38](#_Toc483243779)

[参 考 文 献 39](#_Toc483243780)

[致 谢 40](#_Toc483243781)

# 引 言

随着移动通信、无线传感器网络技术的发展以及普适计算概念的提出，人们对室内定位的需求越来越高，室内环境下基于位置的服务越来越受到人们的关注。室内定位已成为了一个非常活跃的研究领域，并且具有极其广阔的应用前景。众所周知，全球定位系统（GPS）是目前应用最广泛的定位系统，由于其高精度、低成本等优势，在日常生活中被广泛使用。在室外定位的场景中，GPS主要是利用卫星导航根据已知位置的卫星到用户接收机的距离采用三边测量法来确定接收机的具体位置。但是对于某些复杂的室内定位场景，GPS信号很难穿透钢筋混凝土外壁，同时复杂的室内环境导致GPS信号散射、反射现象十分严重，会大大降低GPS的定位精度。对于处于建筑密集，或者情况较为复杂的室内，如大型商场、医院等，GPS便无法达到较好效果且较为耗电。

由于GPS在室内定位中的种种劣势使得采用其他定位方法成为必然趋势。蓝牙规范作为一种短距无线数据与语音传输的开放性全球规范，目前在整个世界范围内都得到了很广泛的应用。它可以支持便携式计算机、移动终端以及其他电子设备之间通过建立无线电空中接口相通信，可以方便地进行数据和语音传输。另一方面，蓝牙所需硬件的成本已下降到了相对较低的水平，且成为一般移动通信终端的标准配置，因此基于蓝牙的定位系统可以有效地利用现有的硬件投入，降低定位的成本并且适合大范围推广用。

因此，可以构建一个基于蓝牙的定位系统，对使用配有蓝牙的手机用户实现定位追定位的室内的RSSI特征信息，并将其放入指纹库；在线阶段安卓手机将测量得到的与周围蓝牙节点的RSSI特征值通过http请求发送到服务器上，服务器通过模糊指纹算法计算出位置信息，也通过http请求发送到安卓手机端，完成定位。

### 1 绪论

## 1.1 研究背景及研究意义

随着社会的发展，不论室内还是室外，人们对于定位的需求都在变得愈来愈迫切，通过当前技术实现定位就成了研究的重点。在室内定位场景中，GPS信号很难穿透墙壁，同时室内场景又十分复杂，会大大降低GPS的定位精度，无法满足需求[1]。

蓝牙技术作为全球普遍支持的短距离数据与语音传输规范，目前广泛使用在各种设备上。能够实现在短距离内的数据交换，相比于Wi-Fi等技术，蓝牙设备的普及率更高，随着蓝牙4.0的推出，其功耗和成本进一步降低。同时蓝牙还提供了许多最基本的参数，比如链路质量（LQ）、信号强度（RSSI）等[2]。正是基于上述这些原因，本文基于蓝牙技术进行定位系统的开发。

## 1.2 研究现状及发展趋势

基于WiFi的定位技术：WiFi芯片在各种手机和移动设备上已经普遍应用，而且其基础热点设施的室内覆盖也非常好，很多需要定位的公共场所如机场、商场都有覆盖，所以WiFi用于室内定位成为很自然的选择，很多现有的解决方案都是主要基于WiFi技术。基于WiFi的定位技术主要有三种，第一种是基于接收信号强度的三边测量定位(接收信号强度定位法)，这也是现在业界应用最多的技术。接收信号强度定位法是通过信号强度和已知信号衰弱模型来估计参考点与待测点的距离，根据多个参考点距离待测点的距离值画出圆，多个圆的重叠部分就是待测目标的位置。它的优点是布局和维护成本相对低，只需要采集WiFi热点的位置数据库，局限是给出的定位精度低，大概能得到10~20m的精度，有些情况可能更低。第二种是基于接收信号强度的指纹定位。该技术是将测量到的接收信号强度与前期测量的各个参考点的信号强度特性进行比较，选取匹配最好的参考点位置来作为测量目标的位置。现有很多解决方案也是专注在该技术。该技术的缺点是布局和维护的成本较高，系统依赖射频信号强度的指纹数据库，对于大规模的使用，数据库大，产生和维护成本相对较高，也在一定程度上造成可移植性差。第三种是基于信号飞行时间的测量，通过测量无线信号在两个节点之间的往返飞行时间，并用该时间推算节点间的距离，根据多个参考点距离待测点的距离值画出圆，多个圆的重叠部分就是待测目标的位置。 它的优点是精度高，可以达到低过1米的精度，其局限是生态系统还不完备，现有的设备不支持。现有的WiFi定位技术都依赖WiFi热点位置分布数据库或者接收信号强度指纹数据库实现定位，所以WiFi定位大范围在室内应用要解决的主要问题是可靠地产生和动态维护数据库，同时要低复杂度，这样才能在大量的建筑中使用。

基于惯性传感器的辅助定位：惯性传感器如加速度计、陀螺仪、磁场计和压力传感器已经被广泛地应用在手机和平板电脑等移动设备上，它们提供9轴或12轴自由度的运动估算。这些传感器可以估计用户的运动状态，实时地推算用户的运动轨迹及高度信息。根据估算的运动轨迹，系统可以给出目标用户相对初始点或参考点的相对位置。它的优点是在相对短的时间内可以给出准确的相对位置估计，并且位置更新速度快，另外功耗也较低;其缺点是长时间的估算误差大，另外有些系统对用户会提出使用限制，比如设备只能采取特定的指向。基于惯性传感器的辅助定位已经开始被广泛应用。

地图匹配技术：室内地图技术也需要迅速发展，地图匹配技术可以帮助定位技术提高精度，比如通过地图拓扑结构分析来修正惯性传感器给出的位置推算结果，可以很好地提高惯性传感器位置推算的精度。

室内定位已经迎来了高速发展的阶段，现在可以看到和预测的趋势可以总结以下几点：

（1） 基于低功耗蓝牙BLE的定位技术会被越来越广泛应用。蓝牙室内定位已发展一段时间，但过去并未受重视，随着苹果公司提出的iBeacon技术， BLE 定位受到相当大的关注。目前基于该技术的信息推送应用在零售业已经获得相当大的响应，预期未来BLE室内定位技术会更多结合信息推送、移动支付等应用，或被实践在日常生活上提供个性化服务。

（2）多种技术结合的混合定位方法，以满足各种室内环境和应用场景的需求，并弥补单一技术的局限。越来越多的解决方案已经将9轴或12轴惯性传感器技术和其它技术如基于WiFi的定位技术融合到一起，并且这已经成为趋势。Google、Broadcom、CSR 都提供多种技术融合的混合解决方案。未来会有更多的解决方案是完整的传感器/WiFi/BLE的混合解决方案以满足多种需求。

（4） 室内地图和室内定位数据库会迅速发展，相关技术趋于成熟，以保证快速扩展的能力和定位性能的可靠性与一致性。这里的挑战包括地图和数据库的扩展性，快速有效地产生和维护数据库的技术，比如通过众包的方法。

（4） 基于位置的应用和服务会更多利用附近的感应和发现。相对定位而言，附近的发现会更简单因为它并不需要计算精确位置，而只是发现附近的设备就能提供相应的服务。这种技术对于室内和连续无处不在的定位而言有很大影响，并可以作为很好的补充，尤其是针对精确定位不容易实现的场景。相关的技术有BT/BLE、LTE Direct、WiFi Direct、NFC等。

（5） 采用专用的定位引擎来处理定位、运动检测、传感器数据分析、信息融合和地理围栏等。通过用专用的处理器来处理运动，情境和定位，可以降低对应用处理器的唤醒，以优化和降低功耗，达到随时随地都知道所处位置的目标。

（6） 低功耗优化，降低定位功能对移动设备带来的额外功耗以实现随时随地的精准定位。包括使用专用的定位处理引擎以尽量少唤醒应用处理器，结合运动检测和行为模式的检测来降低功耗，通过多种定位技术的融合选择最省电同时满足精度的技术，并关闭或使高功耗的定位技术处于休眠模式，以降低高功耗传感器的使用等。

## 1.3 本文研究内容

本论文通过低功耗蓝牙技术和室内定位常用算法进行论述，对蓝牙系统硬件、定位算法、定位软件等进行了研究，实现了一套完整可靠的室内定位系统。

## 1.4 章节安排

第一章：介绍室内定位的研究意义、现状以及室内定位研究的原理和技术。

第二章：介绍常见的定位算法以及其优缺点，分析其定位原理，同时介绍本定位系统中所使用到的技术，比如spring框架、Matlab、低功耗蓝牙的基本性能数据、通信方式、硬件架构等。。

第三章：进行本室内定位系统的需求分析，详细分析系统各部分功能。

第四章：进行整个系统的设计，包括硬件、软件、算法设计。介绍模糊指纹定位算法，对整个定位原理、过程进行详细的叙述。

第五章：介绍本系统的实现方法，结合软件、硬件、算法对整个系统进行详细的实现叙述。包括部分核心代码以及系统实现思想。

第六章：介绍测试的相关理论，同时对整个系统进行完整的测试，最后进行总结。

# 2 蓝牙室内定位相关技术

## 2.1 常见室内定位算法

定位算法通常可分为一下两类，非测距算法和测距算法。

### 2.1.1 非测距算法

1. 近似三角形内点测试法（APIT）

近似三角形内点测试法（APIT）是通过确定待定点与周围所有参考点构成的三角形的关系，找出待定点所在的三角形，计算这些三角形的质心作为待定点位置的方法。当三个锚节点都判断出待测点在远离自身或靠近自身的时候，该锚节点就在这三个锚节点围成的三角形之外；否则，该待测点则在这三个锚节点围成的三角形内。该算法应用场景偏窄，精度较低，无法满足室内定位需求。

1. DV-HOP算法

DV-HOP算法是利用矢量路由和GPS定位思想提出的一系列分布式定位方法之一。

第一阶段：网络中的各参考节点通过典型的距离矢量交换协议向邻居节点广播自身位置信息分组，使得网络中的所有节点获得距参考节点的最小跳数信息。

第二阶段：每个节点通过其它参考节点的位置信息以及相隔最小跳数来计算平均每跳距离，并将其作为一个校正值广播至网络中。当接收到校正值后，节点根据跳数计算与参考节点之间的距离。

第三阶段：当未知节点获得与三个或更多参考节点的距离时，根据三边测量法或极大似然估计法来计算未知节点的位置。

1. 质心算法

利用某些先验的位置信息，比如节点可能存在的某些位置点，对这些位置点采用加权或者最小二乘运算得到节点的位置。

### 2.1.2 基于测距的算法

（1） TOA算法

目前TOA算法最成功的应用就是GPS定位系统。TOA算法是通过测量信号传播时间来估计距离，它的精确性严重依赖于节点间精确的时间同步。GPS系统主要是通过在卫星上搭载高精度的原子钟，以及地面站的定时校准来保证时间同步准确性。系统复杂度和成本都较高。而传统无线传感器网络中又普遍存在节点的同步精度较差的问题，因此TOA算法的定位方案不适合使用基于蓝牙技术的定位系统。

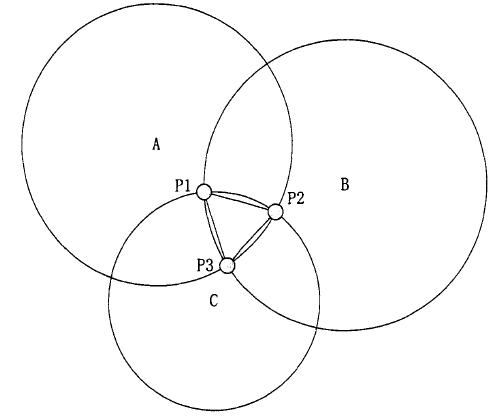
（2） TDOA算法

TDOA算法与TOA算法类似，都是通过测距来实现定位的。它一般是在节点上安装两种无线收发器，工作时利用两种信号在空间传播的速度差异，直接把时间差异转化为距离差。TDOA技术定位精度理论精度可达厘米级，但实际应用中效果不够理想，主要是由于信号传播过程中受干扰影响较大。AOA技术也容易受信号传播的影响，噪声、非视距、多径等问题都会增加定位误差。同时这两种算法都需要额外硬件，无法满足目前成本和功耗的要求。

### 2.1.3 节点定位算法

（1） 三角质心法

定位模型如图2.1所示。三角质心法原理如下：基于信号传输模型，将测出的锚节点和待测点之间的信号强度转换为距离。每个锚节点根据相应的距离作为半径，以自身位置作为圆心做圆，理想状态下，三个圆应该相交于一点，这个点的位置就是待测点的位置。但是由于室内环境复杂，信号的折射、散射十分严重，基于信号传输模型计算出的锚节点和待测点之间的距离并不准确，通常计算出来的距离会比实际距离远。所以，我们可以将三个圆或多个圆相交的区域当做待测点可能出现的区域，再通过取质心的方法计算出待测节点的实际位置。



5.1 三角质心法示例

（2） 加权三角质心算法

加权三角质心算法是在原有三角质心算法上进行改良，具体来说就是加入每个锚节点对定位结果的权重影响。由于每个锚节点距待测点距离不同，所处环境也不尽相同，所以会导致每个锚节点采集的信号强度的精度不同。由于加权质心算法的基本思想是通过加权因子来表现锚节点对质心定位的权值大小，所以加权三角质心算法的关键点就在于如何对锚节点的权值进行界定。

## 2.2 蓝牙概述

蓝牙一词是古挪威语的一个英语化版本，该词是十世纪的丹麦一个国王的绰号，据说该位国王将分裂的丹麦的部落统一为一个强大的王国，并引入了基督教。蓝牙技术的开拓者们使用这个名字意为将混乱的短距离通讯协议统一为全球标准。

蓝牙是一种无线技术标准，可以实现移动设备、固定设备和短距离个人区域网的数据交换（使用2.4—2.485GHz的ISM波段的UHF无线电波）。1994年爱立信公司最早开始研发蓝牙技术，在二十世纪末期，三星、苹果、爱立信等多家跨国巨头公司成立了一个特殊兴趣小组（SIG），目标是提供一种可靠的短距离内的无线数据交换技术，同时也进行蓝牙技术的应用推广。现在这个兴趣小组负责监督蓝牙规范的开发，管理认证项目，并维护商标权益。制造商的设备必须符合蓝牙技术联盟的标准才能以“蓝牙设备”的名义进入市场。蓝牙技术拥有一套专利网络，可发放给符合标准的设备。

截止到目前，蓝牙技术已经发展到4.2版本，它为IOT推出了一些关键性能，是一次硬件更新。但是一些旧有蓝牙硬件也能够获得蓝牙4.2的一些功能，如通过固件实现隐私保护更新。所有的蓝牙标准版本都支持向下兼容，让最新的版本能够覆盖所有旧的版本。随着蓝牙技术的发展，在功耗、传输速度、抗干扰性等性能方面都有了很大的改进。目前蓝牙4.0已经广泛商用，安卓4.0版本及以上版本都已经支持。

在短距离通信、数据交换领域内，目前蓝牙技术已经成为最完善、普及率最广的无线通信协议[3]。

### 2.2.1 蓝牙技术的特点

简单地说，蓝牙是一种短程宽带无线电技术，是实现语音和数据无线传输的全球开放性标准。它使用跳频扩谱（FHSS）、时分多址（TDMA）、码分多址（CDMA）等先进技术，在小范围内建立多种通信与信息系统之间的信息传输。Bluetooth的主要技术特点如下：  
 （1） 工作频段：2.4GHz的工科医（ISM）频段，无需申请许可证。大多数国家使用79个频点，载频为(2402+k)MHz（k=0,1，2…78），载频间隔1MHz。采用TDD时分双工方式。  
 （2） 传输速率：1Mb/s（V2.0以上版本）  
 （3） 调试方式：BT=0.5的GFSK调制，调制指数为0.28-0.35。  
 （4） 采用跳频技术：跳频速率为1600跳/秒，在建链时（包括寻呼和查询）提高为3200跳/秒。蓝牙通过快跳频和短分组技术减少同频干扰，保证传输的可靠性。

（5） 语音调制方式：连续可变斜率增量调制（CVSD，Continuous Variable Slope Delta Modulation），抗衰落性强，即使误码率达到4%，话音质量也可接受。  
 （6） 支持电路交换和分组交换业务：蓝牙支持实时的同步定向联接（SCO链路）和非实时的异步不定向联接（ACL链路），前者主要传送语音等实时性强的信息，后者以数据包为主。语音和数据可以单独或同时传输。蓝牙支持一个异步数据通道，或三个并发的同步话音通道，或同时传送异步数据和同步话音的通道。每个话音通道支持64kbps的同步话音；异步通道支持723.2/57.6kbps的非对称双工通信或433.9kbps的对称全双工通信。  
 （7） 支持点对点及点对多点通信：蓝牙设备按特定方式可组成两种网络：微微网(Piconet)和分布式网络(Scatternet)，其中微微网的建立由两台设备的连接开始，最多可由八台设备组成。在一个微微网中，只有一台为主设备（Master），其它均为从设备（Slave），不同的主从设备对可以采用不同的链接方式，在一次通信中，链接方式也可以任意改变。几个相互独立的微微网以特定方式链接在一起便构成了分布式网络。所有的蓝牙设备都是对等的，所以在蓝牙中没有基站的概念。  
 （8） 工作距离：蓝牙设备分为三个功率等级，分别是：100mW（20dBm）、2.5mW（4dBm）和1mW（0dBm），相应的有效工作范围为：100米、10米和1米。

### 2.2.2 蓝牙技术的应用

蓝牙技术由于实现了可靠稳定的的短距离无线数据交换，所以大大的提高了设备的移动性。蓝牙技术的应用主要应用在一下方面：

（1） 替代了许多传统的有线数据传输方式，比如在智能手机尚未普及的时候，蓝牙功能就已广泛的应用于手机上。人们使用蓝牙进行数据的交换。

（2） 应用到车载设备、便携式的个人移动设备等，便捷了人们的生活。

（3） 由于蓝牙技术支持多个设备组件维维网，最多支持8个节点，其中1个节点作为主节点，其余7个节点作为从节点。根据以上蓝牙的特性可以组建个人的局域网络。

### 2.2.3 蓝牙协议体系结构

蓝牙协协栈的体系结构如图2.2所示。它是由底层硬件模块，中间层和高端应用层三大部分组成。



图 2. 2 蓝牙协议栈体系结构

（1） 蓝牙的底层模块

底层模块是蓝牙技术的核心模块，所有嵌入蓝牙技术的设备都必须包括底层模块。它主要由链路管理层（LMP Link Manager Protocol）、基带层（BB Base Band）和射频（RF Rodio Frequency）组成[4]。其功能是：无线连接层（RF）通过2.4GHZ无需申请的ISM频段，实现数据流的过滤和传输，它主要定义了对工作在此频段的蓝牙接收机应满足的要求；基带层（BB）提供了两种不同的物理链路（同步面向连接链路SCO Synchronous Connection Oriented和异步无连接链路ACL Asynchronous Connection Less），负责跳频和蓝牙数据及信息帧的传输，且对所有类型的数据包提供了不同层次的前向纠错码（FEC Frequency Error Correction）或循环沉余度差错校验（CTC Cyclic Redundancy Check）；LMP层负责两个或多个设备链路的建立和拆除及链路的安全和控制，如鉴权和加密、控制和协商基带包的大小等，它为上层软件模块提供了不同的访问入口。

蓝牙主机控制器接口HCI （Host Controller Interface）由基带控制器、连接管理器、控制和事件寄存器等组成。它是蓝牙协议中软硬件之间的接口，它提供了一个调用下层BB、LM、状态和控制寄存器等硬件的统一命令，上、下两个模块接口之间的消息和数据的传递必须通过HCI的解释才能进行。HCI层以上的协议软件实体运行在主机上，而HCI以下的功能由蓝牙设备完成，二者之间通过传输层进行交互。

（2） 中间协议层

中间协议层由逻辑链路控制与适配协议L2CAP （Logical Link Control and Adaptation Protocol）、服务发现协议 SDP （Service Discovery Protocol）、串口仿真协议或称线缆替换协议 RFCOM 和二进制电话控制协议 TCS （Telephony Control protocol Spectocol）组成[5]。

L2CAP 是蓝牙协议栈的核心组成部分，也是其它协议实现的基础。它位于基带之上，向上层提供面向连接的和无连接的数据服务。它主要完成数据的拆装、服务质量控制，协议的复用、分组的分割和重组（Segmentation And Reassembly）及组提取等功能。L2CAP允许高达64KB的数据分组。

SDP是一个基于客户/服务器结构的协议。它工作在 L2CAP层之上，为上层应用程序提供一种机制来发现可用的服务及其属性，而服务的属性包括服务的类型及该服务所需的机制或协议信息。

RFCOMM 是一个仿真有线链路的无线数据仿真协议，符合ETSI 标准的 TS 07.10串口仿真协议。它在蓝牙基带上仿真RS-232的控制和数据信号，为原先使用串行连接的上层业务提供传送能力。

TCS是一个基于 ITU-T Q.931 建议的采用面向比特的协议，它定义了用于蓝牙设备之间建立语音和数据呼叫的控制信令（Call Control Signalling），并负责处理蓝牙设备组的移动管理过程。

（3） 高端应用层

高端应用层位于蓝牙协议栈的最上部分。一个完整的蓝牙协议栈按其功能又可划分为四层：核心协议层（BB、LMP、LCAP、SDP）、线缆替换协议层（RFCOMM）、电话控制协议层（TCS-BIN）、选用协议层（PPP、TCP、TP、UDP、OBEX、IrMC、WAP、WAE）。而高端应用层就是由选用协议层组成。

选用协议层中的 PPP （Point-to-Point Protocol）是点到点协议，它由封装、链路控制协议、网络控制协议组成，它定义了串行点到点链路应当如何传输因特网协议数据，它主要用于LAN接入、拨号网络及传真等应用规范；

TCP/IP(传输控制协议/网络层协议)、UDP（User Datagram Protocol 对象交换协议）是三种已有的协议，它定义了因特网与网络相关的通信及其他类型计算机设备和外围设备之间的通信。蓝牙采用或共享这些已有的协议去实现与连接因特网的设备的通信，这样，既可提高效率，又可在一定程度上保证蓝牙技术和其它通信技术的互操作性；

OBEX（Object Exchange Protocol）是对象交换协议，它支持设备间的数据交换，采用客户/服务器模式提供与 HTTP（超文本传输协议）相同的基本功能。该协议作为一个开放性标准还定义了可用于交换的电子商务卡、个人日程表、消息、和便条等格式；WAP（Wireless Application Protocol）是无线应用协议，它的目的是要在数字蜂窝电话和其它小型无线设备上实现因特网业务。它支持移动电话浏览网页、收取电子邮件和其它基于因特网的协议。

蓝牙高层协议栈的主要目标是支持确定目标的用户模型；支持多种硬件平台；具有良好的用户使用接口，能够使用传统的应用，在可能的情况下使用现 有的协议；最小的源码数量；最大的重用；具有可维护性。蓝牙高层协议栈在实现操作系统独立性时使用了虚拟操作系统（VOS Virtual Operating System）的概念，在操作系统和协议之间设计了一个虚拟操作系统接口，把蓝牙主机栈直接用到很多嵌入式实时操作系统中（如：OSE、Psos+、VxWorks 等），同时亦可用到标准的操作系统中（如：Windows 和 UNIX 等），而这些移植和使用都不需要改变协议栈的源码，这样，既提高了效率又在一定程度上保证了不同技术和设备的互操作性；为了实现硬件的独立性，蓝牙协议栈在 HCI 层以下附加了一个串行接口层，实现到不同 HCI 接口的映射；为了利用传统的应用，以及便于使更多的用户能够开发基于蓝牙的应用，蓝牙协议栈提供了蓝牙应用程序接口（API Application Programming Interface）；为了管理和配置协议栈的各个层次，在蓝牙协议栈软件中附加了一个协议栈的管理模块。这样蓝牙协议栈在上述几个方面都实现了有效的隔离，消除了对外部环境的依赖性，用户不再需要对蓝牙协议栈的源码进行任何改变，用户只需通过简单的编译和链接就可以集成进最新的蓝牙主机栈。

## 2.3 spring框架概述

Spring是一个开源的轻量级Java SE（Java 标准版本）/Java EE（Java 企业版本）开发应用框架，其目的是用于简化企业级应用程序开发。应用程序是由一组相互协作的对象组成。而在传统应用程序开发中，一个完整的应用是由一组相互协作的对象组成。所以开发一个应用除了要开发业务逻辑之外，最多的是关注如何使这些对象协作来完成所需功能，而且要低耦合、高内聚。业务逻辑开发是不可避免的，那如果有个框架出来帮我们来创建对象及管理这些对象之间的依赖关系。Spring框架通过配置方式来创建对象，管理对象之间依赖关系，我们不需要通过工厂和生成器来创建及管理对象之间的依赖关系，减少了我们的许多工作，加速了开发，能节省出很多时间来干其他事。

Spring框架除了帮我们管理对象及其依赖关系，还提供像通用日志记录、性能统计、安全控制、异常处理等面向切面的能力，还能帮助管理最头疼的数据库事务，其本身提供了一套简单的JDBC访问实现，提供与第三方数据访问框架集成（如Hibernate、JPA），与各种Java EE技术整合（如Java Mail、任务调度等等），提供一套自己的web层框架Spring MVC、而且还能非常简单的与第三方web框架集成。从这里我们可以认为Spring是一个超级粘合平台，除了自己提供功能外，还提供粘合其他技术和框架的能力，从而使我们可以更自由的选择到底使用什么技术进行开发。而且不管是JAVA SE（C/S架构）应用程序还是JAVA EE（B/S架构）应用程序都可以使用这个平台进行开发。

传统程序开发，创建对象及组装对象间依赖关系由我们在程序内部进行控制，这样会加大各个对象间的耦合，如果我们要修改对象间的依赖关系就必须修改源代码，重新编译、部署；而如果采用Spring，则由Spring根据配置文件来进行创建及组装对象间依赖关系，只需要改配置文件即可，无需重新编译。所以，Spring能帮我们根据配置文件创建及组装对象之间的依赖关系。

当我们要进行一些日志记录、权限控制、性能统计等时，在传统应用程序当中我们可能在需要的对象或方法中进行，而且比如权限控制、性能统计大部分是重复的，这样代码中就存在大量重复代码，即使有人说我把通用部分提取出来，那必然存在调用还是存在重复，像性能统计我们可能只是在必要时才进行，在诊断完毕后要删除这些代码；还有日志记录，比如记录一些方法访问日志、数据访问日志等等，这些都会渗透到各个要访问方法中；还有权限控制，必须在方法执行开始进行审核。如果采用Spring，这些日志记录、权限控制、性能统计从业务逻辑中分离出来，通过Spring支持的面向切面编程，在需要这些功能的地方动态添加这些功能，无需渗透到各个需要的方法或对象中；采用Spring 面向切面编程，通过配置方式，不需要在现有代码中添加任何额外代码，使得现有代码专注业务逻辑。所以，Spring 面向切面编程能帮助我们无耦合的实现日志记录，性能统计，安全控制。

在传统应用程序当中，我们需要获取连接，执行SQL，提交或回滚事务，关闭连接，来完成数据库事务管理。而且还要保证在最后一定要关闭连接。如果采用Spring，我们只需获取连接，执行SQL，其他的都交给Spring来管理了。Spring能非常简单的帮我们管理数据库事务。Spring还提供了与第三方数据访问框架（如Hibernate、JPA）无缝集成，而且自己也提供了一套JDBC访问模板，来方便数据库访问。

Spring提供与第三方Web（如Struts、JSF）框架无缝集成，而且自己也提供了一套Spring MVC框架，来方便web层搭建。同时Spring能方便的与Java EE（如Java Mail、任务调度）整合，与更多技术整合（比如缓存框架）。

## 2.4 Matlab概述

Matlab名字来源于“矩阵实验室”缩写，是美国MathWorks公司推出的一套以矩阵计算为基础的数学软件。它提供了丰富可靠的矩阵运算、图形绘制、数据处理等功能，可以实现数值分析、优化、统计、微分方程求解、信号处理、图像处理等的计算和图形显示，可解决工程、科学计算和数学学科中许多问题。因此，被广泛应用于自动控制、图像信号处理、生物医学工程、语音处理、信号分析、时间序列分析与建模、优化设计等领域。

Matlab功能强大、简单易学、编程效率高，几乎可以完成所有的数学建模的编程需要，是数学建模必备软件。

Matlab可以将函数文件打包为jar包，可以使用java进行调用。通过这种方法可以将复杂的算法逻辑用Matlab实现，而java可以专注于业务逻辑，使整个项目架构更加清晰明了易于维护。

# 3 需求分析

室内定位的算法选用基于指纹技术的室内定位算法。基于指纹技术的蓝牙室内定位系统是通过对蓝牙接收的信号强度（RSSI）的实际测量，建立位置指纹库，再将需要定位的蓝牙设备实测数据与位置指纹数据库进行对比来确定出蓝牙设备的实际位置。室内定位的精度主要与定位算法和蓝牙锚节点有关。

为了更加清晰的说明整个系统的需求，首先介绍整个定位算法的流程。整个定位流程可以分为两个阶段，分别是离线阶段和在线阶段。

离线阶段负责测量定位区域内的蓝牙信号强度信息，并将其建立为指纹库，如图3.1所示。



在线阶段负责实际定位时，客户端将测量到的待测点与蓝牙锚节点之间的信号强度上传到服务器上，同时接收服务器回传的定位信息，如图3.2所示。



图3.1.2在线 阶段

## 3.1 假定和约束

假定本项目的所有用户都有一定的Android操作系统的操作经验，能够熟练使用各种APP应用。使用的移动端设备的操作系统应该在Android4.0以上。使用采集数据APP的用户应该有一定的定位算法基础。

## 3.2 功能需求

### 3.2.1 软件需求

软件需求可以分为提供定位服务的服务器端需求和移动端的需求。

（1） 服务器端需求

顾名思义，服务器端的需求应该是为其它模块提供服务的部分，在蓝牙定位项目中，服务器端应该负责从移动端接收数据，然后进行定位，并把定位结果返回给移动端。总得来说，服务器端应该实现算法，并且要实现网络数据传输的功能。服务器应考虑到并发处理，尤其要考虑性能，每次用户的定位请求的响应时间应尽量缩短。

（2） 移动端需求

需要这款APP有良好的移动设备适配性与兼容性，可以部署在任何一款Android移动终端上，真正实现移动危机处理。而且此款APP要做到简洁明了，便于理解。要求专业美观，具有亲和力，风格与内容相一致。能够提高系统的交互性和可用性，提高用户体验效果。移动端需求可以分为两部分，可以开发两个APP应用进行实现。具体如下：

①数据收集APP

数据收集APP的功能应该是收集一个区域内的锚节点的信号强度信息，服务器端将该信号强度信息用作特征值参与定位。所以数据收集APP的需求是将周围蓝牙锚节点的信号强度测量出来，并将这些信号强度信息以某种方式（比如存放在文件或者写入数据库中）保存下来。

② 定位APP

定位APP的功能是将周围蓝牙锚节点的信号强度信息测量出来后，发送给服务器端，然后从服务器端接收定位结果的数据。所以定位APP的需求是测量周围蓝牙锚节点的信号强度、把信号强度信息发送给服务器端，同时等待接收服务器端返回的定位结果信息。

### 3.2.2 硬件需求

由上文可知，不论数据收集APP还是定位APP，都需要用到蓝牙锚节点，所以硬件的需求是设计实现一种稳定可靠，易于测量出与待测点的信号强度的蓝牙模块。同时该蓝牙模块应该易于部署，操作方便，工作时间持久。

## 3.3 未来可扩展需求

以上是初步的需求，未来可能提出的需求包括但不局限于以下部分：

（1） 安全性：室内定位系统的使用对象为个人，个人的位置信息包含大量个人隐私，假使在使用过程中，蓝牙信号被恶意拦截，将会造成隐私的泄露，影响用户的人身安全。为了保障信息的安全，设计加密算法以提高用户的安全性十分必要。

（2） 定位反馈：由于室内环境复杂，假使某一蓝牙参考由于某种原因，RSSI出现剧烈波动，会对定位精度产生很大影响。为了在出现类似情况出现时及时解决，会引入用户反馈机制，从用户端反馈定位偏差较大的情况，辅助系统提高定位精度。

（3） 减小误差：在收集蓝牙锚节点的阶段如果数据波动较大，为了保证数据的真实性和稳定性，应该考虑使用滤波算法对收集到的信息进行处理，比如使用卡尔曼滤波等方法。

# 4 定位系统设计

## 4.1 硬件设计

计划使用RF-BM-S01低功耗蓝牙模块作为定位蓝牙锚节点。该低功耗蓝牙模块采用TI的CC2541作为核心处理器。模块运行在 2.4 GHz ISM band，GFSK 调制方式（高斯频移键控），40 频道2 MHz 的通道间隙，3 个固定的广播通道，37 个自适应自动跳频数据通道，物理层可以和经典蓝牙RF组合成双模设备，2 MHz 间隙能更好地防止相邻频道的干扰。宽输出功率调节(-23 dBm～4dBm)，-93 dBm高增益接收灵敏度。

该模块的主要优势为：无需任何蓝牙协议栈及射频应用经验；支持桥接模式（透传）和直驱模式（免MCU）；空旷传输距离100m；高速传输塑料最高可达6Kbp/S超低功耗通信TX Current≤0.5mA RX Current≤0.5mA；超低待机功耗0.3-0.4uA;支持与苹果4s、iPad3、iPod4以后的机型通信；支持与蓝牙4.0功能的安卓android4.3以上手机通信。

## 4.2 算法设计

### 4.2.1 位置指纹

室内环境复杂，存在墙壁、桌椅等障碍物，会影响到室内的信号传播；同时，室内的移动物体比如人体、车辆等也会对蓝牙信号的传播产生干扰，再加上信号传播本身的衍射以及多径效应，类似TOA、TDOA等算法在复杂的室内环境中会收到很大干扰。

由于室内特殊的环境，导致在室内的每个位置上的信号强度会展示出不同的特性以及特征值。这事由于不同位置的信道多径结构是互不相同的，信号在传播过程中，会因传播途径上不同位置上互不相同的多径结构，产生和位置有密切关系的特征。由此产生了基于位置指纹的定位算法[6]。该算法的主要过程是先预先采集待定位区域的蓝牙信号指纹信息，然后在服务器建立与之对应的指纹库。当需要定位的时候，将待测点的蓝牙信号信息传给服务器，服务器依据之前采集的指纹库进行比对估算，最后完成待测点的定位。这种算法能较好的适应复杂的室内环境[7]。

### 4.2.2 加权K近邻算法

简单的说，加权K近邻算法采用测量不同特征值之间的距离方法进行分类。

加权K近邻算法，它的工作原理是：存在一个样本数据集合，也称作训练样本集，并且样本集中每个数据都存在标签，即知道每个数据与所属分类对应关系。输入没有标签新数据后，将新数据的特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。

一般来说，我们只选择样本数据集中前K个最相似的数据，这就是k-近邻算法中k的出处，通常k是不大于20的整数。最后选择k个最相似的数据中出现次数最多的数据，作为新数据的分类。

### 4.2.3 模糊C均值聚类划分

模糊C均值聚类算法是一种基于划分的聚类算法，它的思想就是使得被划分到同一簇的对象之间相似度最大，而不同簇之间的相似度最小[8]。

模糊C均值是普通C均值聚类算法的改进，普通C均值对数据进行硬性划分，一个样本一定明确的属于某一类，模糊C均值算法对数据进行模糊划分，使用隶属度表示一个样本属于某一类的程度。实际聚类中可能会遇到这样的情况，蝴蝶形数据集中样本点的类别不好硬性判断，所以引入隶属度来进行模糊划分[9]。

模糊C均值聚类算法的具体步骤如下：

初始化：给定聚类类别数C，而且C满足条件

 (4.1)

其中n为数据个数，设定迭代停止阈值ε，初始化聚类原型模式，将迭代计数器设置为b=0；

步骤一：用公式计算或更新划分矩阵：

对于i,k，如果，则有：

 （4.3）

如果i,r，使得，则有：

 （4.3）

步骤二：用公式更新聚类原型模式矩阵：

 （4.4）

其中i=1,2...c

步骤三：如果满足条件：

 （4.5）

则算法停止并输出划分矩阵U和聚类原型P，否则令

 （4.6）

并转向步骤一。

在对高维矩阵的处理方面，使用模糊聚类可以方便找出异常点，减少因多重计算而带来的开销，具有一定的优势。指纹库中采集到的数据，经过模糊聚类算法处理之后，可以生成模糊相似矩阵，相似矩阵可以作为以后操作的对象，避免了多次重复遍历数据库。通过对定位区域的划分，也可消除一些误差较大的点的影响，提高算法的精度。

## 4.3 软件设计

服务器端和移动端之间的通信方式使用http协议，原因如下：

（1） http协议简单快速，客户向服务器请求服务时，只需传送请求方法和路径，请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同；

（2） 由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快；HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记，非常灵活；

（3） http是一种无连接的通信方式，无连接的含义是限制每次连接只处理一个请求，服务器处理完客户的请求，并收到客户的应答后，即断开连接，采用这种方式可以节省传输时间。

服务器端和移动端约定使用使用成熟的JSON格式进行数据交换，原因如下：

（1） 由于JSON格式是压缩的，所以占用带宽小；

（2） 对于JSON格式的数据，不论是服务器端还是移动端，JSON都可以十分方便快捷的进行解析。

### 4.3.1 服务器端设计

服务器使用IntelliJ IEDA工具开发，使用java语言，为了开发更加简便，同时为了整个项目低耦合、高内聚，业务代码集中于业务逻辑上，所以使用SpringMvc框架。服务器端主要设计为两个接口，分别是在线阶段定位接口和更新指纹信息接口。

（1） 在线阶段定位接口

在线阶段定位接口的流程如图4.1所示。首先移动端调用该接口同时传入待测点的信号强度信息，判断内存中是否有预先采集好的指纹信息，如果没有则从文件中加载到内存中，然后将待测点信号强度信息和指纹信息传入算法jar包中，然后将获得的定位信息回传给移动端，整个服务器端的流程结束。考虑到java调用Matlab的jar包用时会很长但有不可避免，所以为了提高该定位接口的响应速度，便将指纹信息提前加载到内存中，如果内存中存在指纹信息，响应速度会大大提高。同时，使用这种提前将数据加载到内存中的方法，数据持久化的方案则选择了文件存放。为了保证热更新内存中的指纹信息，又考虑设计了将指纹信息从文件加载到内存中的接口，该接口将在下文进行详细叙述。



图4.1 服务器端定位流程图

（2） 更新指纹信息接口

该接口主要面向服务器端管理人员。设计更新指纹信息接口的目的是不用停掉服务器的服务就可以更新内存中的指纹信息。具体来说就是规定使用http请求调用该接口，服务器端接收到该请求后便重新将文件中的指纹信息加载到内存中。同时为了安全性问题，服务器端和移动端会约定一个特殊的请求头字段，防止该接口被恶意的反复调用，影响服务器端性能。

### 4.3.2 移动端设计

根据定位流程可以将移动端部分分为两个部分，分别是离线阶段采集蓝牙锚节点的指纹信息的APP和在线定位阶段测量待测点位置信息的定位APP。由于RF-BM-S01低功耗蓝牙模块需要操作系统android4.3及以上的平台支持，所以移动端平台选用android4.3以上的系统平台。使用eclipse进行安卓平台的APP开发。

（1） 采集指纹信息APP

该APP面向参与前期待定位区域指纹信息采集的定位工作人员。工作流程为：首先打开手机蓝牙；然后在需要测量指纹信息的位置上扫描周围布置的蓝牙锚节点，测量得到该收集与周围各个蓝牙锚节点之间的信号强度信息，APP会把每个待测点的指纹信息按照和服务器端约定好的格式保存到文件中；最后将该保存指纹信息的文件进行整理，留作服务器端在线定位阶段使用。

（2） 在线定位APP

该APP面向需要定位服务的普通用户，定位流程为：首先打开手机蓝牙，然后测量周围蓝牙锚节点的信号强度信息，然后通过http协议将这些信号强度信息发送给服务器端；最后接收服务器端回传的位置信息，移动端使用直观的方式，比如在地图上标记目前所在位置等方式，展示最终的定位结果。

# 5 定位系统实现

## 5.1 硬件实现

RF-BM-S01 蓝牙模块是基于TI 公司CC2540 芯片研发的低功耗蓝牙（BLE）射频模块，可广泛应用于短距离无线通信领域。具有功耗低、体积小、传输距离远、抗干扰能力强等特点。模块配备高性能蛇形天线；模块采用半孔形式硬件接口设计。该模块可用于开发基于蓝牙4.0（BLE，低功耗蓝牙）的消费类电子产品，手机外设产品等，能提高操作的可靠性；提高信号的传输距离和抗干扰性；还能实现解决不同电子产品间的互操作问题，电池寿命也可显著延长。

主从模式可切换。模块可以工作在桥接模式 (透传模式 )和直驱模式。模块启动后会自动进行广播，已打开特定APP的手机会对其进行扫描和对接，成功之后便可以通过BLE协议对其进行监控[10]。在本系统中将该蓝牙模块设置为从模式状态下，安卓手机在主模式状态下工作，有安卓手机的蓝牙模块发起扫描周围蓝牙锚节点。

桥接模式下，CPU可以通过模块的通用串口和移动设备进行双向通讯，也可以通过特定的串口AT指令，对某些通讯参数进行管理控制。数据的具体含义由上层应用程序自行定义。移动设备可以通过APP对模块进行写操作，写入的数据将通过串口发送给用户的CPU。模块收到来自CPU串口的数据包后，将自动转发给移动设备。此模式下的开发，必须负责主CPU的代码设计，以及智能移动设备端APP代码设计。

直驱模式下，用户对模块进行简单外围扩展，APP通过BLE协议直接对模块进行驱动，完成智能移动设备对模块的监管和控制。此模式下的软件开发，负责智能移动设备端APP代码设计。

蓝牙模块实物图如图5.1所示。

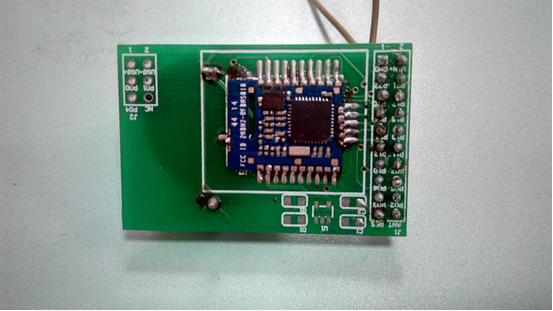


图 5.1 蓝牙模块实物图

## 5.2 算法实现

### 5.2.1 位置指纹

该算法的主要过程是先预先采集待定位区域的蓝牙信号指纹信息，然后在服务器建立与之对应的指纹库。所以预先采集的待定位区域的蓝牙信号强度信息就是定位算法中的位置指纹信息。

### 5.2.2 加权K近邻算法

在获取理想的位置参考节点之后，要使用加权KNN算法进行位置估算。在模糊C均值聚类算法中，求出了不同参考定对每一子区域的隶属度，假如使用模糊隶属度作为权重进行位置估算，当在某一单独区域内选取参考点时，此时每一点的权重代表的是该参考点对该子区域的隶属关系，于待定位点的位置信息没有很大的关联，即某一点对该类的隶属度高并不代表着它和带定位点的距离很近，也有距离很远的可能，假如单纯地依据模糊隶属度为各个参考点分配权重，则估算出的位置具有一定的随机性。而且在同一子区域中的很多位置进行位置估算时，有可能选取同样的参考点，在这种情况下不同位置所对应的估算值是一样的，显然不符合定位的要求。如果计算待定位点对某一区域的模糊隶属度，又会带来巨大的时间开销，而定位算法对实时性有较高的要求。因而在这种情况下，使用模糊隶属度作为权重是不合适的[11]。

为每一点分配权重的原则是：给离待定位点距离近的点分配较大的权重，随着距离的增加，分配的权重数值也随之变小。依据这个原则，最简便有效的方法便是采用距离作为权重。即不同点的权重分配为：

 （5.1）

其中为第i个待定位点和第j个参考点之间的欧氏距离，为一个很小的数，目的是防止距离为零时运算失去意义，在本文研究的算法中其取值为0.001。当在不同的区域内取值时，待定位点已经被明确地划归为某一类，选取的参考点中有属于另外一类的点，在这种情况下可以使用模糊隶属度来作为衡量权重的一个因素[12]。此时权重的分配为：

 （5.2）

式中表示参考点j对子区域k的模糊隶属度。在不同情况下选用不同的权重分配原则，对算法定位准确性的提高也有一定的帮助。

在matlab环境下算法的实现代码如下：

for i = 1:4

c(i,:) = a-center(i,:)； %求待定位点与类中心点距离矩阵

end

[y,h] = sort (sum(abs(c),2))；

D\_value = y(2) - y(1)； %取距离最小的两个值作差，判断待定位点在类边缘的情况

%当待定位点不在类边缘时

temp = indexsize (h(1))；

disp (temp);

temp1 = zeros ( h(1),1)；

indextemp = zeros(1,temp)；

locatemp = size(temp,2)；

for k1 = 1:h(1)

temp1 (k1,1) = indexsize (k1)；

end

for w =1:temp

indextemp(w) = indexs (sum(temp1,1)-temp+w)；

end

locatemp =loca (indextemp,:)； %提取定位点所在区域的蓝牙参考点信息

wew = repmat ( a , size(b,1),1)；

v=b-wew；

D = sum ( abs ( v ) , 2 )；

[z , n] = sort ( D )； %求待定位点与所在区域内参考点的距离矩阵

for k = 1:5

Indexx ( k , 1) = n ( k )；

H (1,k) = 1 / ( z ( k) + 0.1 )； %计算定位参考点权重

end

locaX = locatemp ( indexx, : )； %提取定位参考点坐标

当定位点处于两类边缘时代码实现流程如上，不同的部分是由于取点时要从两个区域内选取，所以需要两次提取定位区域内的蓝牙参考点信息。在提取这些信息时，同时要将隶属度信息提取出来。通过对实验结果的查看分析，存在一些点即是类边缘的点，同时也是空间边缘的点，在对这些点进行定位时，会先从两类中选取5个定位参考点，之后选取前两个参考点进行位置估算，但是由于选取的参考点的依据并不是与待定位点之间的距离，所以会出现较大的误差。为了解决这个问题，在算法中对选取的K个参考点根据距离的关系重新排序。下面为实现这些功能的主要代码：

U1 = U (h(1),:); %提取蓝牙参考节点的隶属度

U2 = U1 (1,indextemp1)；

for k2=1:3

Utemp1 = U2 (1,i(k2))；

end

U3 = U1 (1, indextemp )；

for k4 = 1:2

Utemp2 = U3 (1,i(k4))；

end

Uloca = [ Utemp1, Utemp2]；

[z2,i2] = sort (indexx4)； %对定位点进行排序并计算权重

for k5 = 1:5

H (1,k5) = Uloca (1,i2(k5)) / (z2(k5)+0.1)；

end

%对待定位点进行位置估算

E(j,1) = H\*locaX(:,1)/temp；

E(j,2) = H\*locaX(:,2)/temp；

dis = sqrt (((locax(j,1)-E(j,1)).^2) + ((locax(j,2)-E(j,2)).^2))；

%对空间边缘的待定位点进行处理

if ( dis>0.9 )

for m=1:2

tempH (1,m) = H(1,m)；

templocax (m,:) = locaX(m,:)；

end

E(j,1) = tempH \* templocax (:,1) / sum (tempH)；

E(j,2) = tempH \* templocax (:,2) / sum (tempH)；

dis = sqrt (((locax(j,1)-E(j,1)).^2) + ((locax(j,2)-E(j,2)).^2))；

end

### 5.2.3 模糊C均值聚类划分

在基于位置指纹的算法中，离线阶段采集蓝牙参考节点发送的数据，建立指纹数据库。并采用模糊C均值聚类算法对指纹数据库进行聚类划分。算法中各参数的设置分别为：最大迭代次数为100，隶属度矩阵U的指数取值为2，隶属度最小变化量取0.00001。以下为模糊C均值聚类算法的matlab实现。

① 初始化模糊C均值聚类算法的隶属度函数矩阵。

其中U为初始化的隶属度矩阵，cluster\_n为聚类中心个数，data\_n为样本点数。

U = rand (cluster\_n, data\_n) ;

col\_sum = sum (U) ;

U = U. / col\_sum (ones(cluster\_n, 1), :) ;

② 计算新的隶属度矩阵。

mf = U.^expo ; % 隶属度矩阵进行指数运算结果

center = mf \* data. / ((ones (size(data, 2), 1) \* sum(mf'))'); % 新聚类中心

dist = distfcm (center, data) ; % 计算距离矩阵

obj\_fcn = sum(sum((dist.^2).\*mf)) ; % 计算目标函数值

tmp = dist.^(-2/(expo-1)) ;

U\_new = tmp./(ones(cluster\_n, 1)\*sum(tmp)) ; % 计算新的隶属度矩阵

其中计算距离矩阵的实现如下：

out = zeros(size(center, 1), size(data, 1)) ;

% 对每一个聚类中心 每一次循环求 得所有样本点到一个聚类中心的距离

for k = 1:size (center, 1)

out (k, :) = sqrt ( sum ((( data-ones (size (data,1) , 1) \* center (k,:)).^2)',1)) ;

end

③ 迭代计算。先进行条件判断，如果相邻两次目标函数值差的绝对值小于阈值，则终止迭代，输出聚类结果。否则继续迭代直至满足条件为止或者迭代次数达到100次。代码如下：

if i > 1

if abs (obj\_fcn(i) - obj\_fcn(i-1)) < min\_impro ；

break;

end

end

使用模糊C聚类对指纹数据库进行划分后，可以将指纹库划分为若干类。本算法设定的划分类别数为4。在划分结束后可以提取每一类的位置指纹信息以及相对应的位置信息，供定位阶段使用。提取每类指纹信息以及位置信息的代码如下所示：

maxU = max (U) %求隶属度矩阵中每一行的最大值，并作为矩阵maxU输出

index1 = find (U(1,:) == maxU) ; %第一类所对应的蓝牙参考点序数

index2 = find (U(2,:) == maxU) ; %第二类所对应的蓝牙参考点序数

index3 = find (U(3,:) == maxU) ; %第三类所对应的蓝牙参考点序数

index4 = find (U(4,:) == maxU) ; %第四类所对应的蓝牙参考点序数

loca1 = loca (index1,:) ； %第一类的蓝牙参考点位置

loca2 = loca (index2,:) ； %第二类的蓝牙参考点位置

loca3 = loca (index3,:) ； %第三类的蓝牙参考点位置

loca4 = loca (index4,:) ； %第四类的蓝牙参考点位置

## 5.3 软件实现

### 5.3.1 移动端实现

移动端实现如图5. 2所示，图左是采集指纹APP，图右是在线定位APP。

图5.2 移动端实现

（1） 采集指纹信息APP

因为扫描周围蓝牙节点的过程很长，如果写在安卓主线程里，会因为响应时间过长导致系统主动关闭程序，所以必须将执行时间过长的扫描周围蓝牙节点的过程写在一个线程里，我使用实现Runnable接口的run方法来新开辟一个线程；扫描周围蓝牙节点信息是采集指纹信息APP的核心部分，实现如下：

① 先判断蓝牙是否打开：

if (mBluetoothAdapter.isEnabled())

② 蓝牙如果是打开状态的，则进行扫描：

mBluetoothAdapter.startDiscovery();

③ 进行蓝牙设备的发现：

String action = intent.getAction();

if (BluetoothDevice.ACTION\_FOUND.equals(action))

④ 获得蓝牙设备名：

BluetoothDevice device = intent.getParcelableExtra(EXTRA\_DEVICE);

String name=device.getName();

⑤ 如果设备名预置的蓝牙锚节点的名称一致，则测量信号强度并写到文件中：

short rssi = intent.getExtras().getShort(BluetoothDevice.EXTRA\_RSSI);

File sdCardDir = Environment.getExternalStorageDirectory();

File saveFile = new File(sdCardDir, "result.txt");

FileWriter outStream = new FileWriter(saveFile, true);

outStream.write(rssi);

outStream.close();

（2） 在线定位APP

扫描周围蓝牙节点信号强度的方法和上文类似，所以这里就不赘述。在线定位APP需要把待测点和周围蓝牙定位锚节点的信号强度传给服务器端；因为http请求部分如果写在主线程里，会导致主线程阻塞，最终安卓系统会主动关闭应用，所以我将http请求部分单独写在一个线程里，使用实现Runnable这接口的run方法来实现。发送http请求的具体实现如下：

① 创建代表请求的httpCient对象：

HttpClient httpCient = new DefaultHttpClient();

HttpGet httpGet = new HttpGet(url);

② 执行请求，获取服务器发还的相应对象：

HttpResponse httpResponse = httpCient.execute(httpGet);

③ 检查相应的状态是否正常，若检查状态码的值是200则表示正常：

if (httpResponse.getStatusLine().getStatusCode() == 200)

④ 从相应对象当中取出数据并处理：

HttpEntity entity = httpResponse.getEntity();

String response = EntityUtils.toString(entity,"utf-8");：

⑤ 处理完成后发出通知：

Message message = new Message();

message.obj = response.toString();

showmessage\_handler.sendMessage(message);

为了方便用户浏览定位结果，所以将定位结果用图形的方式表现出来，因为要等服务器返回定位结果后才能画图，画图用时也过长，所以也要将画图功能写在线程里，同时这也是一个异步的处理，提高了APP的响应灵敏性。具体实现为继承View类重写其中的onDraw方法，核心代码如下：

① 创建用来接收请求的Handler对象：

Handler showmessage\_handler = new Handler();

② 获取服务器发还的相应对象并进行处理：

JSONObject result = new JSONObject((String) msg.obj);

String response = (String) msg.obj;

③ 在开辟的线程中调用更新画板函数：

myView.handler.sendEmptyMessage(0x1234);

④ 画板内进行最终的定位结果图形化表示：

super.onDraw(canvas);

Paint paint = new Paint();

canvas.drawText(XLabel[i], XPoint + i \* XScale, YPoint + 50, paint);

### 5.3.2 服务器端实现

服务器端类图如图5. 3所示。BaseController负责监听请求，根据URL将不同的请求转发给不同的函数；然后会调用BaseService，进行实际的业务逻辑的执行，BaseService会调用文件工具类FileUtil；BaseController根据BaseService执行后返回的结果判断执行是否成功，然后调用ResultCode将与移动端约定好的执行成功或执行失败的返回码返回给移动端。



图5.3 服务器端类图

（1） 在线阶段定位接口

BaseController中的getPosition函数监听/getPosition的请求，请求到来后，BaseController会调用BaseService中的getPosition函数。这里展示最关键的BaseService中的getPosition函数的主要实现代码：

① 如果指纹信息为空，则从文件中加载到内存中：

if(null == double\_a1 || null == double\_ax1 || null == double\_loca1 || null == double\_locax1)

reload();

② 调用Matlab的算法包，传入指纹信息和待测点蓝牙信号强度信息：

Object[] input = {MapLoca,TestLoca,MapXY,TestXY};

Object[] back = matlab.locationx2(4, input);

③ 通过工具类FileUtil处理Matlab的算法包返回的定位结果，并将其放在结果中返回给BaseController：

double[] double\_back = FileUtil.getArrayFromString(backArray.toString());

result.put("code", ResultCode.SUCCESS\_CODE);

result.put("xPoint", double\_back[0]);result.put("yPoint", double\_back[1]);

return result;

④ BaseController将结果返回给移动端：

return baseService.getPosition(param1\_double, param2\_double);

（2） 更新指纹信息接口

① 如果http请求头中的字段不等于之前约定好的值则会返回执行失败：

if(!xtoken\_value.equals(xtoken)){

JSONObject result = new JSONObject();

result.put("code", ResultCode.FAIL\_CODE);

return result;

}

② BaseController调用BaseService的reload函数，将指纹信息加载到内存中：

double[][] double\_ax1\_1 = FileUtil.readFromFile();

（3） 使用Matlab将算法部分打成jar包，如图5. 4所示。

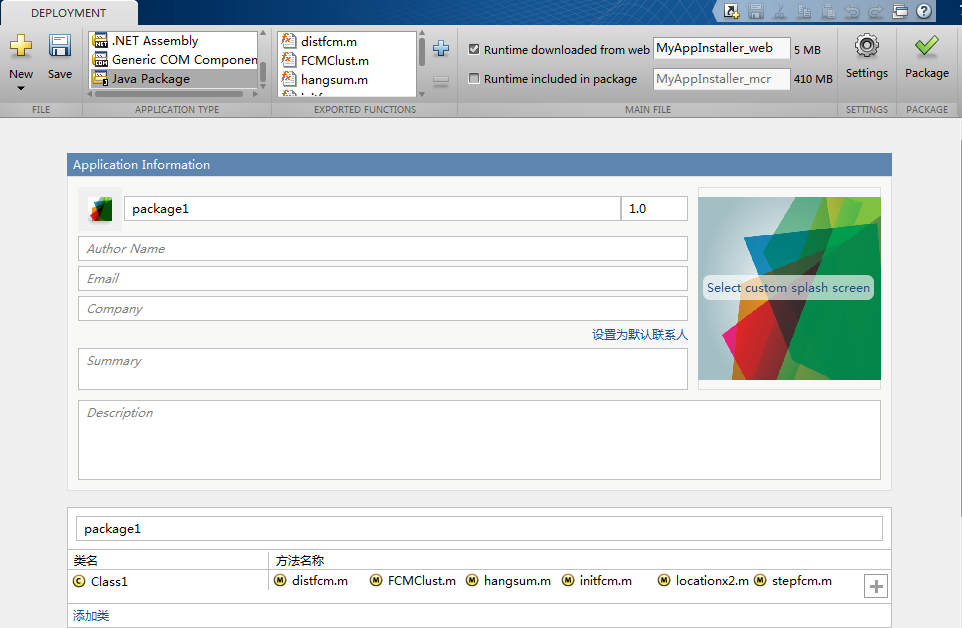


图5.4 算法打包

需要注意的是，Matlab的java版本要确保和本机的java环境版本保持一致。

# 6 定位系统测试

## 6.1 测试理论

软件测试起源于上世纪70年代中期，在1975年软件测试才被确定为一种研究方向；在1979年Glen Ford Myers的《软件测试的艺术》（The Art of Software Testing）给软件测试定义为：测试是为了发现错误而执行的一个程序或者系统的过程。

软件测试就是：使用人工或自动手段来运行或测试某一系统的过程，其目的在于发现错误，检验是否满足用户需求或弄清预期结果与实际结果的差别。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。是把测试对象看作一个黑盒子。利用黑盒测试法进行动态测试时，需要测试软件产品的功能，不需测试软件产品的内部结构和处理过程。采用黑盒测试设计测试用例的方法有：等价类划分、边界值分析、错误推测、因果图和综合策略。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范，是否美观，是否安全，做某个性能测试的时候，首先要保证它的功能正确，然后再考虑该功能点的性能测试。

## 6.2 测试结果

### 6.2.1 硬件测试

该蓝牙的名字是Tc221u-5EA68404，在安卓手机上可以搜索到，如图6.1所示。



图6.1 手机搜索到蓝牙模块

使用数据收集APP可以采集到该蓝牙的名称以及与测试手机之间的信号强度，如图6.2所示。



图6.2 采集数据APP测量到的蓝牙模块

可以测量出该蓝牙模块的名称、相对与测试手机之间的信号强度，表明该蓝牙模块可以满足本系统的基本使用需求。

### 6.2.2 软件测试

（1） 服务器端测试

我使用功能强大的工具Postman来测试服务器端接口。Postman是一款功能强大的网页调试与发送网页HTTP请求的Chrome浏览器插件。相对于其他客户端软件如rest client， 在线的postman更加的方便、强大。

① 定位接口测试：

服务器端首先测试的是最终要的定位接口getPosition接口。请求方式选择get方式，在url内传入待测点周围的蓝牙节点的信号强度作为参数传给服务器，如图6.3所示。返回的结果为JSON对象；code代表执行结果，如果code等于20000，则表示执行成功，其它值则为表示服务器内部执行错误；xPoint和yPoint表示在地图上的位置信息。

使用Postman还可以看到每次接口响应请求的耗时，如图6.3所示，本次测试耗时1686毫秒。经过多次的测试后发现该接口的响应时间基本稳定在1300毫秒作用，性能方面的需求也已经基本满足。

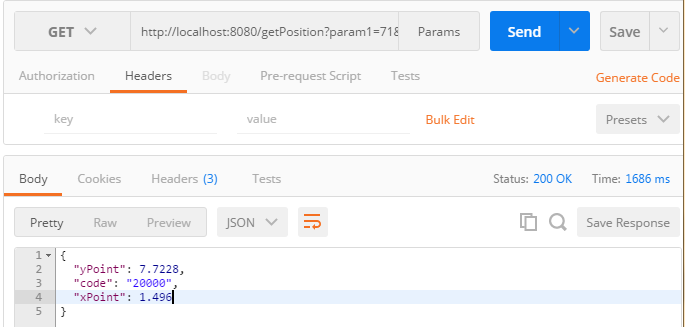


图6.3 定位接口测试结果

② 重新加载指纹信息接口：

为防止该接口被恶意反复调用，从而影响服务器端性能，每次请求需要在请求头中加入xtoken参数，测试结果如下所示。首先是不传xtoken参数，测试结果如图6.4所示。

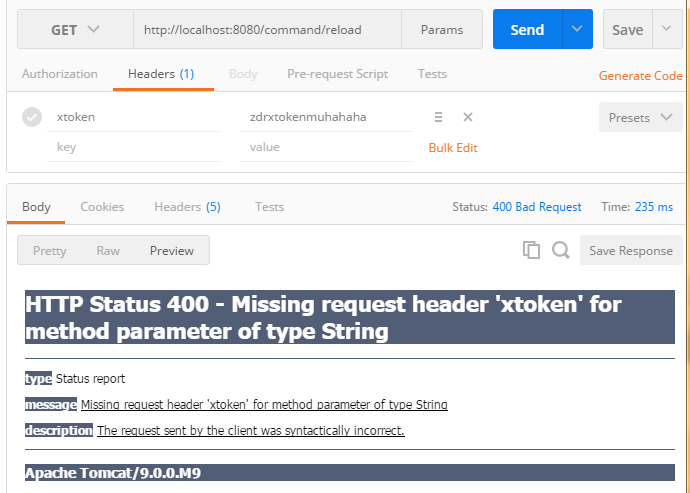


图6.4 不传token调用重新加载指纹信息接口

然后是传入错误的token值，请求方式选择get方式，测试结果如图6.5所示。返回值code是20001表示传入token值不正确，测试通过。

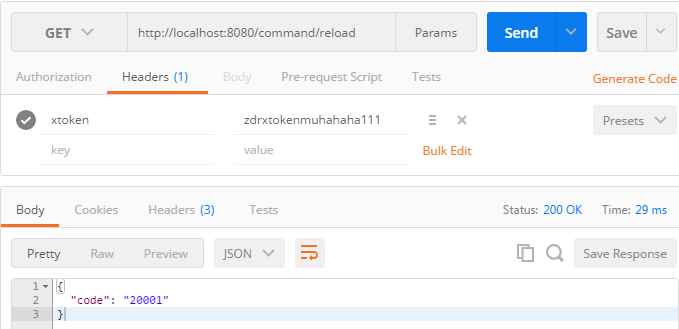


图6.5 传入错误token调用重新加载指纹信息接口

最后是正确调用重新加载指纹信息接口，如图6.6所示。经过多次测试该接口耗时基本稳定在290毫秒左右，满足设计时的需求。综上所述通过测试。

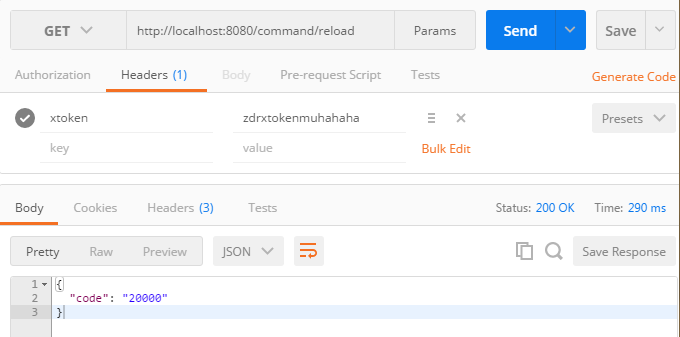


图6.6 正确调用重新加载指纹信息接口

（2） 移动端测试

① 采集指纹APP测试

采集指纹APP可以测量出蓝牙锚节点与测试手机之间的信号强度，如图6.2所示。同时，将指纹信息保存到文件的测试结果如图6.7所示。测试过程中没有出现APP没有响应或者闪退等问题，可以打开或者关闭蓝牙，正常的测量信号强度并记录到文件中，满足设计需求，通过测试。

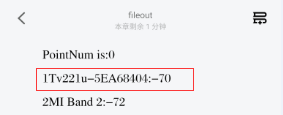


图6.7 指纹信息保存测试

② 在线定位APP测试

首先进行蓝牙信号强度的测试，点击测量按钮。如图6.8所示。

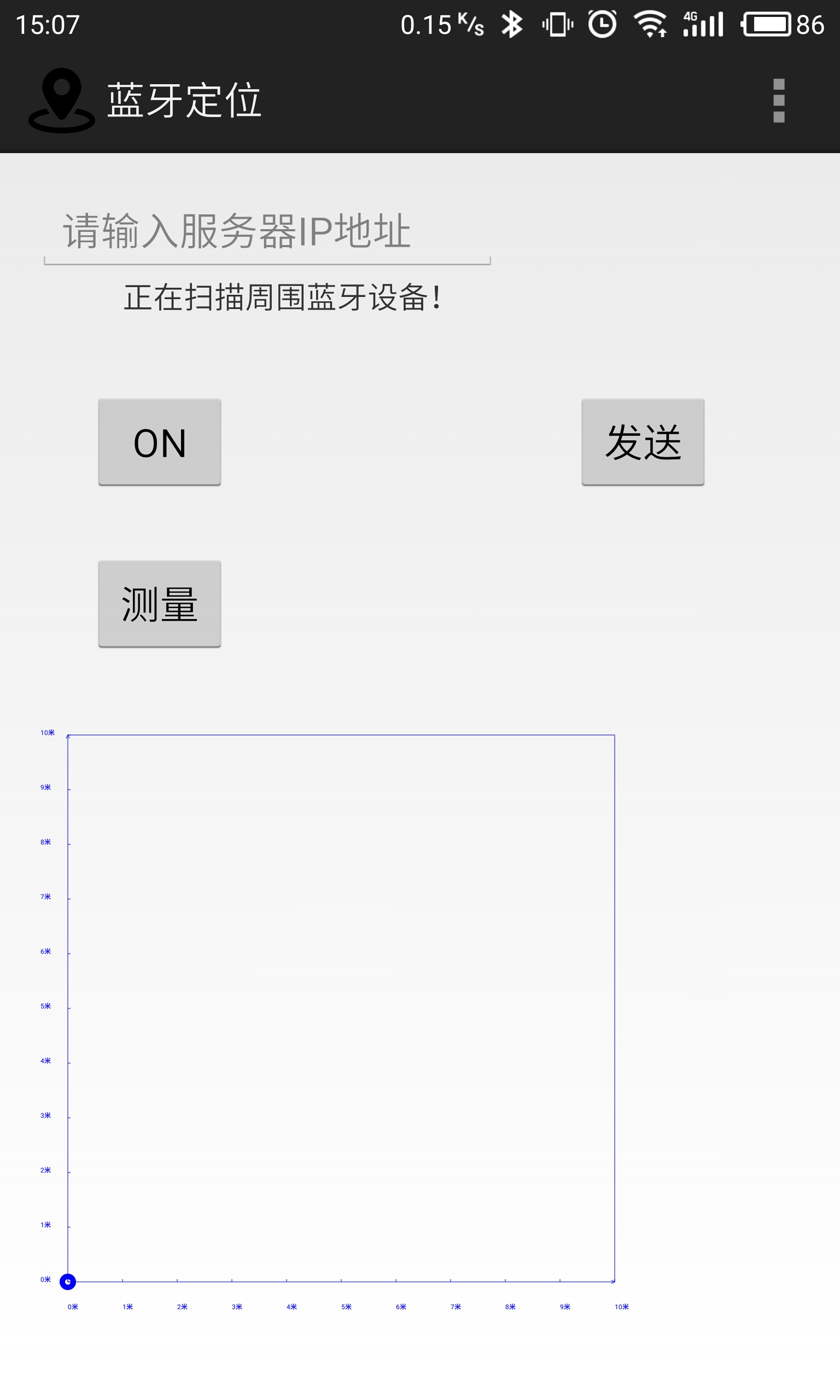


图6.8 测量蓝牙信号强度测试

测量完成后如图6.9所示。测量开始时APP状态提示框为正在扫描周围蓝牙设备，扫描完成后该提示框变成over，满足设计需求，通过测试。

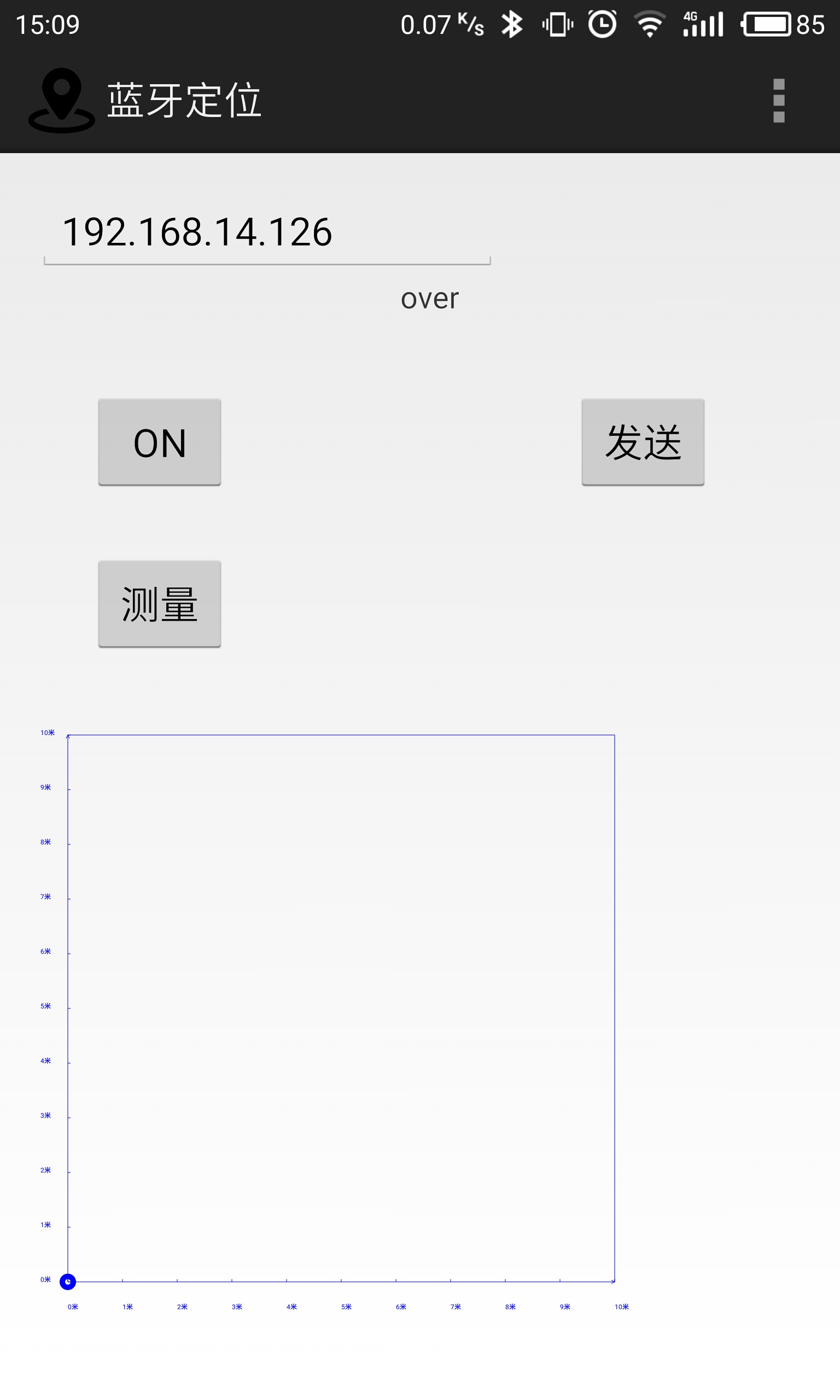


图6.9 测量完成

测量完成后输入正确的服务器IP地址，点击发送按钮，如图6.10所示。服务器端接收到了在线定位APP发送的参数，服务器端执行完毕后，在线定位APP接收到了服务器端回传的定位结果。在线定位APP调用自身的画图线程，将位置信息画在画板上，如图6.10靠近Y轴的蓝色圆点。

面向用户的在线定位APP在测试过程中没有出现闪退或没有响应等情况，能够通过http请求向服务器端发送请求，也能够将服务器端返回的定位数据展示在画板上，满足了设计要求，通过测试。

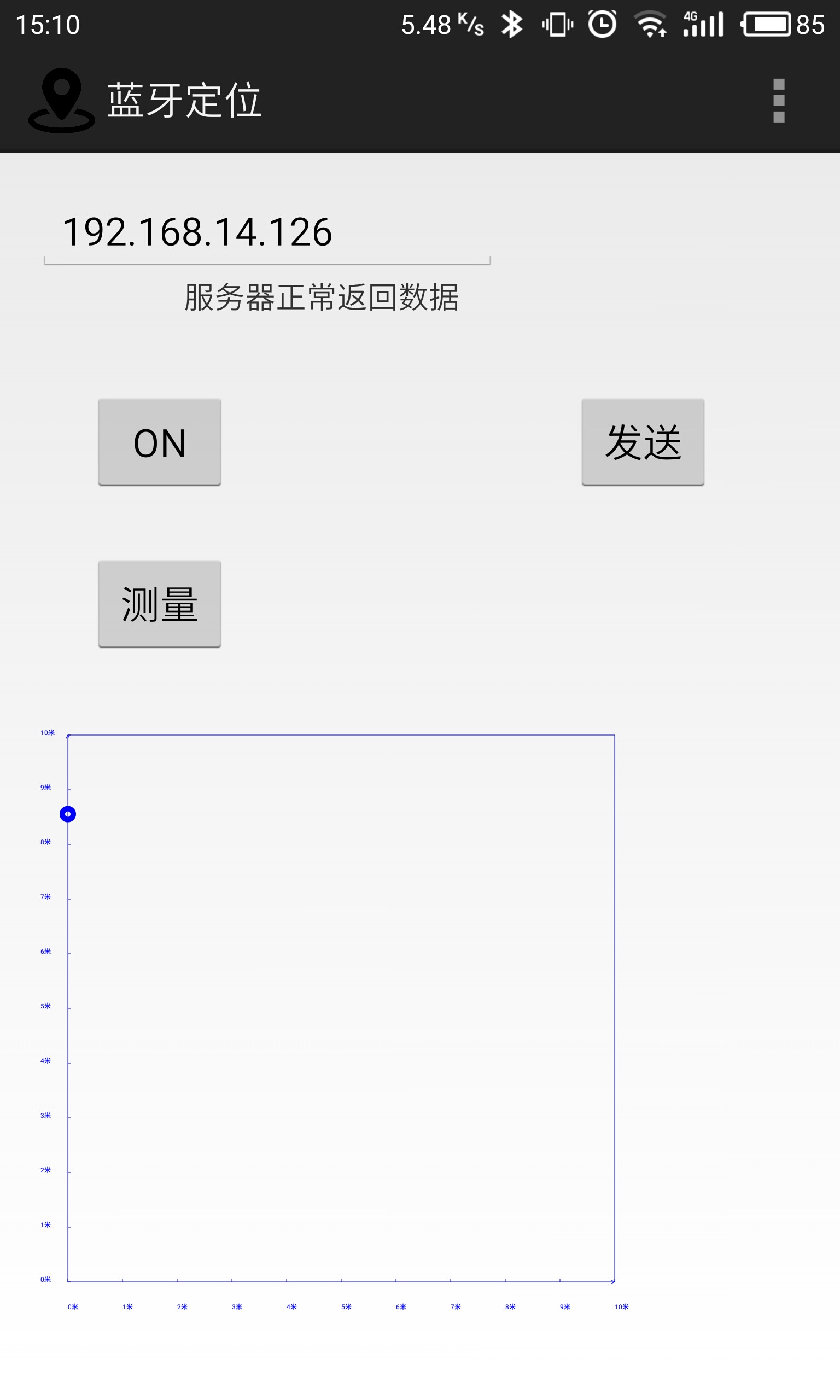


图6.10 定位完成

（3） 算法测试

定位参考点的选取即选择合适的参考点，作为后续流程的输入。在选取参考点时，本算法将数据之间的距离大小作为选取的指标。不同数据之间的距离运算公式为：

 （6.1）

根据公式6.1生成仿真的指纹库。首先先使用理想状态下的指纹数据和定位待测点数据。为了验证本文算法的实验精度，在空间中选取了60个待定位点。考虑到蓝牙参考节点的分布特性，在空间的四周区域中参考点部署少。根据待定位点出现在这些区域内的概率，合理地安排 了60个参考点的构成。在60个待定位点中，有20个参考点出现在空间四周参考点部署很少的区域，其它40个点分布的周围都分布有较多的蓝牙参考点。Matlab运行的仿真结果如图6.11所示，图中圆点表示待测点的实际位置，十字表示定位算法的运行结果。通过统获得定位误差的均值为0.51米，使用仿真数据可以获得不错的定位效果。

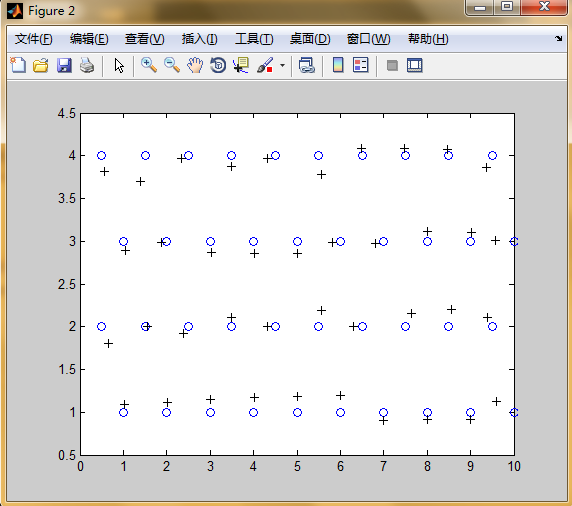


图 6.11 使用仿真数据的定位结果

测量完成后输入正确的服务器IP地址，点击发送按钮，如图6.10所示。服务器端接收到了在线定位APP发送的参数，服务器端执行完毕后，在线定位APP接收到了服务器端回传的定位结果。在线定位APP调用自身的画图线程，将位置信息画在画板上，如图6.10靠近Y轴的蓝色圆点。

表4.1 网络训练参数设置

|  |  |
| --- | --- |
| 参数名称 | 参数值 |
| net | CIFAR-10/train\_test.prototxt |
| test\_iter | 100 |
| test\_interval | 2000 |
| test\_initialization | false |
| base\_lr | 0.001 |
| momentum | 0.9 |
| weight\_decay | 0.004 |
| lr\_policy | multistep |
| gamma | 0.1 |
| stepvalue | 60000 |
| stepvalue | 65000 |
| display | 10000 |
| average\_loss | 100 |
| max\_iter | 200000 |
| snapshot | 10000 |
| snapshot\_prefix | CIFAR-10/cifar10 |
| solver\_mode | GPU |

测量完成后输入正确的服务器IP地址，点击发送按钮，如图6.10所示。服务器端接收到了在线定位APP发送的参数，服务器端执行完毕后，在线定位APP接收到了服务器端回传的定位结果。在线定位APP调用自身的画图线程，将位置信息画在画板上，如图6.10靠近Y轴的蓝色圆点。

表4.1.1 网络训练参数设置

|  |  |
| --- | --- |
| 参数名称 | 参数值 |
| net | CIFAR-10/train\_test.prototxt |
| test\_iter | 100 |
| test\_interval | 2000 |
| test\_initialization | false |
| base\_lr | 0.001 |
| momentum | 0.9 |
| weight\_decay | 0.004 |
| lr\_policy | multistep |
| gamma | 0.1 |
| stepvalue | 60000 |
| stepvalue | 65000 |
| display | 10000 |
| average\_loss | 100 |
| max\_iter | 200000 |
| snapshot | 10000 |
| snapshot\_prefix | CIFAR-10/cifar10 |
| solver\_mode | GPU |

测量完成后输入正确的服务器IP地址，点击发送按钮，如图6.10所示。服务器端接收到了在线定位APP发送的参数，服务器端执行完毕后，在线定位APP接收到了服务器端回传的定位结果。在线定位APP调用自身的画图线程，将位置信息画在画板上，如图6.10靠近Y轴的蓝色圆点。

表4.1 网络训练参数设置

|  |  |
| --- | --- |
| 参数名称 | 参数值 |
| net | CIFAR-10/train\_test.prototxt |
| test\_iter | 100 |
| test\_interval | 2000 |
| test\_initialization | false |
| base\_lr | 0.001 |
| momentum | 0.9 |
| weight\_decay | 0.004 |
| lr\_policy | multistep |
| gamma | 0.1 |
| stepvalue | 60000 |
| stepvalue | 65000 |
| display | 10000 |
| average\_loss | 100 |
| max\_iter | 200000 |
| snapshot | 10000 |
| snapshot\_prefix | CIFAR-10/cifar10 |
| solver\_mode | GPU |

测量完成后输入正确的服务器IP地址，点击发送按钮，如图6.10所示。服务器端接收到了在线定位APP发送的参数，服务器端执行完毕后，在线定位APP接收到了服务器端回传的定位结果。在线定位APP调用自身的画图线程，将位置信息画在画板上，如图6.10靠近Y轴的蓝色圆点。

表5.1 拉伸的激发了三顿饭

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

本系统由传感器节点与基站组成，

# 结 论

本文基于蓝牙4.0技术，使用模糊指纹算法，依托于安卓平台，实现了一个整套的低功耗蓝牙室内定位系统，该系统主要应用于复杂的室内环境中。经过实验的验证，定位平均误差为0.51米，基本满足了室内定位的需求。

该系统的硬件系统基础为低功耗的RF-BM-S01蓝牙4.0模块，该蓝牙模块价格低廉、功耗较低、技术成熟；同时目前觉多数的安卓手机都支持蓝牙4.0，可以安装本系统的在线定位APP，所以在系统的推广上具有很大的优势。

本系统服务器端使用SpringMvc框架，使得系统的可维护性大大提高，方便未来的服务器功能升级，为以后系统的维护升级打下基础；服务器端运行时采用将指纹库加载进内存的方式，大幅减少了运行过程中数据访问的时间消耗，使得服务器端响应请求的时间大大缩小。移动端和服务器端使用http请求进行交互，减少了服务器端响应时间；数据交换的格式统一使用JSON数据格式，提高了代码的可读性。

算法方面将采集到的蓝牙信号强度作为特征数据，使用基于位置指纹的定位算法，同时使用模糊C均值聚类算法进一步优化，最终实现待测点的定位。基于位置指纹的定位算法在室内环境下有良好的定位精度，能够较好的适应复杂的室内环境。

在整个系统的实现过程中仍存在一些不足，需要在未来的工作中进一步完善。未来应尽快解决移动端的界面美化优化问题；同时如果指纹库数据量过大，服务器端使用其它易于维护的方式来保存指纹库（比如使用mysql数据库等）；算法方面，定位空间中的某些特殊位置定位偏差较大应做特殊处理。本系统的实现为室内位置信息的获取提供了有效的解决方法。

参 考 文 献

[1] 马耀辉.基于指纹技术的蓝牙室内定位系统研究[D].重庆邮电大学,2013.

[2] Zhu, Jianyong. RSSI based Bluetooth low energy indoor positioning[C]. International Conference on Indoor Positioning and Indoor Navigation IEEE, 2015:526-533.

[3] 徐金苟.低能耗蓝牙4.0协议原理与实现方法[J].微型电脑应用,2012,28(10):16-19.

[4] 李方.基于ZigBee的位置指纹法室内定位技术研究[D].哈尔滨工业大学,2012.

[5] 张会清,石晓伟,邓贵华,等.基于BP神经网络和泰勒级数的室内定位算法研究[J].电子学报,2012,40(9):1876-1879.

[6] 张明华,张申生,曹健.无线局域网中基于信号强度的室内定位[J].计算机科学,2007,34(6):68-71.

[7] 曹喆.基于低功耗蓝牙和位置指纹的室内定位系统的研究与实现[D].云南大学,2014.

[8] 范霄.基于WiFi信号的无线室内定位[D].山东科技大学,2013.

[9] Kaemarungsi K, Koweerawong C, Wipusitwarakun K. Indoor localization improvement via adaptive RSS fingerprinting database[C]. International Conference on Information NETWORKING. IEEE, 2013:412-416.

[10] Forno F, Malnati G, Portelli G. Design and implementation of a Bluetooth ad hoc network for indoor positioning[J]. 2005, 152(5):223-228.

[11] Zou D Y, Sun H, Chen Y, et al. A Research on Positioning Technology Based on Internet of Things[J]. Applied Mechanics & Materials, 2013, 303-306:926-929.

[12] Mi Q, Stankovic J A, Stoleru R. Practical and secure localization and key distribution for wireless sensor networks [J]. Ad Hoc Networks, 2012, 10(6):946-961.

# 致 谢

时光荏苒，岁月如梭，四年大学时光转眼已接近尾声。在这里要首先感谢我的毕业设计指导老师，在毕业设计期间，老师为我提供了良好的学习环境和条件，也给予了悉心的指导。老师严谨的治学态度和求实创新的科研精神让我受益匪浅，正是因为老师的无私帮助，我才得以顺利完成论文。

于此同时，还要感谢实验室里同组的同学，他们的勤奋和踏实深深地影响了我，让我更有动力去鞭策自己。感谢李岩同学在蓝牙硬件上给我的帮助，感谢武帆同学在信号仿真和测试上给我的帮助。

同时感谢教过我的老师和班级的同学给予过我的帮助，还有家人给我的支持，在以后的学习和生活中，我希望能用更好的成果来答谢他们。

最后，向参加论文评阅和答辩的所有老师表示衷心的感谢！