

# IndoorPos

---

这是一个采用iBeacon技术的室内定位服务端程序，里面包含了三种定位算法，三边定位算法、加权三边定位算法和三角形加权质心定位算法。程序采用Spring框架，简化了JDBC和RMI的样板式代码，并配置了Druid数据库连接池，同时，为了满足高并发的需求，采用了NIO框架Netty，更好地支持大量终端同时定位。程序已经Maven化。

## 一、室内定位系统简介

---

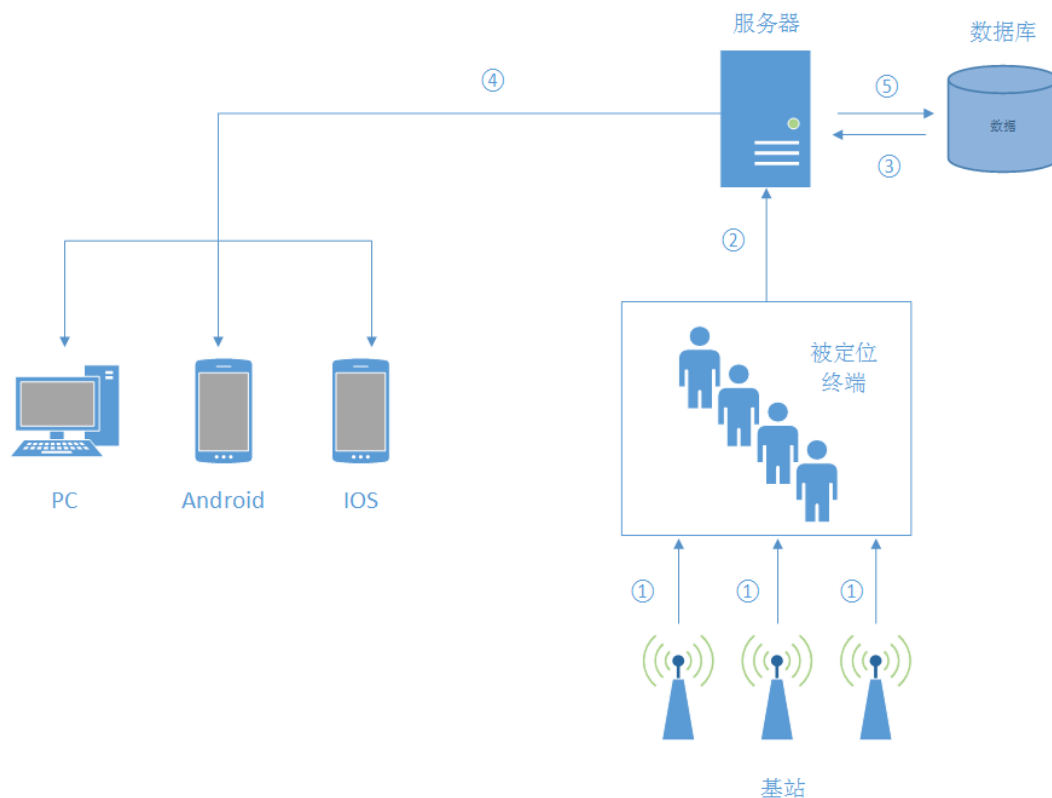
由于卫星信号到达地面时较弱、不能穿透建筑物等问题，在室内环境中无法使用卫星定位。本室内定位系统是一种采用蓝牙4.0 iBeacon基站定位技术，结合高容错定位算法进行室内定位的系统，定位基站廉价、易安置，定位终端可直接采用支持蓝牙4.0的移动终端，由于目前绝大多数移动终端都已支持蓝牙4.0，因此本系统不需要使用额外的定位标签、设备，直接可以通过移动电子设备进行定位。从业务应用的角度来看，本系统可实现对人员的实时定位、导航指路等功能以及贵重物品、车辆的定位；同时还可进行区域人流分析，人员密度分析，停留时间、人员流量统计，特定人员监控，区域报警和人员轨迹查看等多种应用的二次开发。

室内定位系统可分为基站、被定位终端、服务器、数据库、客户端等五个部分，分别详述如下：

1. 基站：布置于需要定位的环境，向外发射蓝牙信号，各基站发射频率与发射强度保持一致，每一均有独立ID及相对于定位环境坐标原点的坐标，基站向外发射包括自己ID的信号，基站坐标保存在数据库中。
2. 被定位终端：所有具备蓝牙4.0功能的设备均可作为被定位终端，其接收基站发来的信号强度值与基站的ID信息，并连同该被定位终端的ID信息，一同发送到服务器中。
3. 服务器：服务器接收从被定位终端发来的该定位终端ID信息、接收到信号的基站ID信息及对应的信号强度值，并从数据库中取得基站坐标信息，利用定位算法进行定位，得到被定位终端的坐标信息，先发送给客户端，然后写入到数据库。
4. 数据库：存储定位的相关数据，包括高度补偿值、基站坐标、衰减因子值、员工信息，并保存服务器发来的被定位终端的坐标信息。
5. 客户端：客户端分为PC端、IOS端、Android端，负责接收服务器发来的坐标信息，然后进行前端展示。

系统的架构图如下：

室内定位系统架构图



图中各流程注解如下：

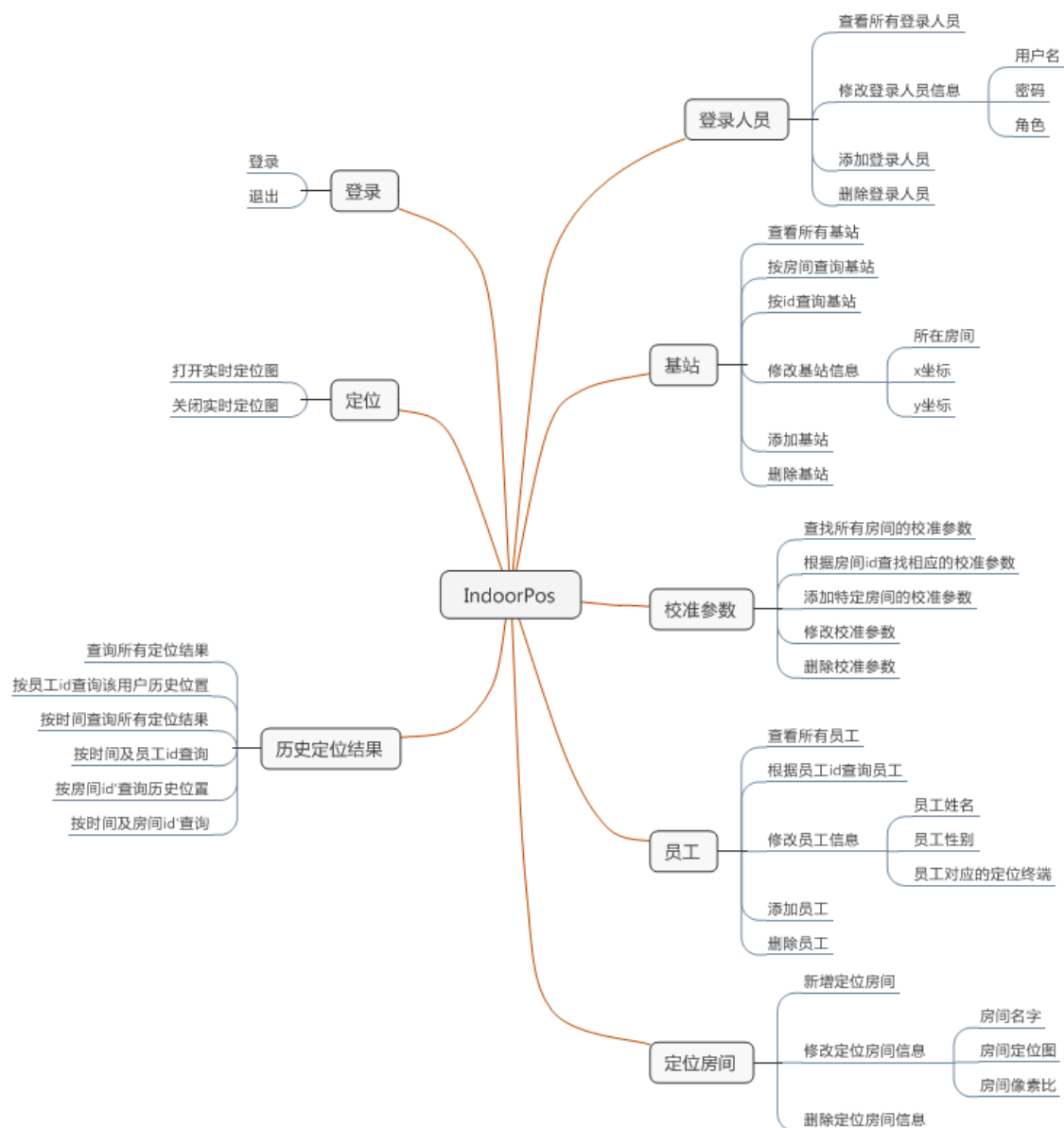
- ①：基站向需要定位的环境发送包括自身ID、强度值的信号。
- ②：被定位终端向服务器发送自身ID、接收到信号的基站ID组及其对应的信号强度值
- ③：服务器从数据库中拿到高度补偿值、基站坐标、衰减因子值用以定位。
- ④：服务器向客户端发送被定位终端的ID及坐标信息。
- ⑤：服务器向数据库中存储时间戳、被定位终端ID及其对应的坐标信息。

## 二、本程序简介

本程序属于室内定位系统的服务器端程序,主要功能有两大块：

1. 负责接收被定位终端发来的信息并利用定位算法计算定位结果，将定位结果发送给显示客户端进行显示并将定位结果保存到数据库；
2. 通过发布RMI服务，与客户端进行通信，完成室内定位的相关管理工作。

本程序的功能结构图如图所示：



## 三、定位算法简介

本系统采用的是蓝牙4.0 iBeacon定位技术，其原理是定位终端接收到iBeacon基站发来的信号强度，然后根据无线信号强度的渐变模型得出基站与被定位终端的直线距离，然后再根据高度补偿法，得出基站与终端的平面距离，当终端接收到三个以上不同基站的信号，即能得出与三个以上不同基站的水平距离，且这些基站的坐标坐标已知，就可以对这个终端进行定位。

### 3.1建模与测量终端到基站的距离

实际中通常用来测得基站与终端距离的简化无线信号渐变模型如下：

$$[p(d)]_{\text{dBm}} = [p(d_0)]_{\text{dBm}} - 10n \lg\left(\frac{d}{d_0}\right)$$

式中， $P(d)$ 表示距离基站直线距离为 $d$ 时终端接收到的信号强度，即RSSI值； $P_0(d)$ 表示距离基站为 $d$ 时终端接收到的信号功率； $d_0$ 为参考距离，为计算方便，通常选择一米处为参考距离； $n$ 是路径损耗(Pass Loss)指数，通常是由实际测量得到，障碍物越多， $n$ 值越大，从而接收到的平均能量下降的速度会随着距离的增加而变得越来越快。

在实际应用中，通常需要实地测量得到基站在一米处接收到的功率值、环境衰减因子、高度补偿三个值，分别记为 $p_0$ 、 $n$ 、 $h$ 。其中， $h$ 根据终端一般使用时，与基站的垂直距离得到； $p_0$ 、 $n$ 测量时，由于具体室内模型的建立与优化，对定位效果影响最大，为使模型能够最大程度符合当前室内环境中的无线信号传播特性，使RSSI测距能获得更高的精度，需要对参数 $A$ 和 $n$ 进行优化进而得到当前室内环境下的最优值。一般通过线性回归分析来估计参数 $A$ 和 $n$ 的值，因为RSSI值在超过14m以后基本趋于平缓，不再符合接收信号强度随着距离增大而衰减的规律。所以为保证测距精度，基站固定后，以20 cm为间隔，在距离基站14m的范围内设置70个测量点，即距离基站0.2 m, 0.4 m, ..., 14m等位置。在每个测试点接收100个数据包后，对100个RSSI值求平均值，再以平均后的RSSI值作为终端在该位置收到的信号强度。最后记录RSSI和 $d$ 的对应关系，这样就得到了70组测量数据 $(\rho_i, \text{RSSI}_i)$ ,  $i = 1, 2, 3, \dots, 70$ ，其中 $\rho_i$ 表示距离为 $d_i$ 时终端接收到的功率值。对所采集的70组测量数据使用线性回归分析，带入以下公式，即可求出 $p_0$ 、 $n$ （式中 $A$ 表示 $p_0$ ）：

$$\rho_i = -10 \lg d_i, i = 1, 2, \dots, 70$$

$$n = \frac{\sum_{i=1}^{70} (\rho_i - \bar{\rho}) \text{RSSI}_i}{\sum_{i=1}^{70} (\rho_i - \bar{\rho})^2}$$

$$A = \overline{\text{RSSI}} - n\bar{\rho}$$

其中

$$\bar{\rho} = \frac{1}{70} \sum_{i=1}^{70} \rho_i$$

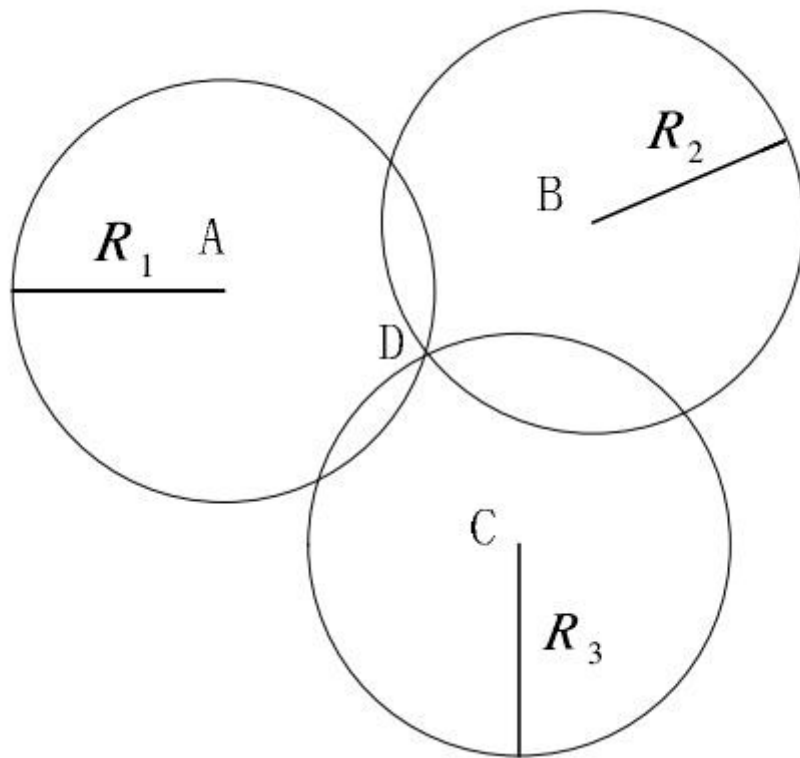
$$\overline{\text{RSSI}} = \frac{1}{70} \sum_{i=1}^{70} \text{RSSI}_i$$

## 3.2定位

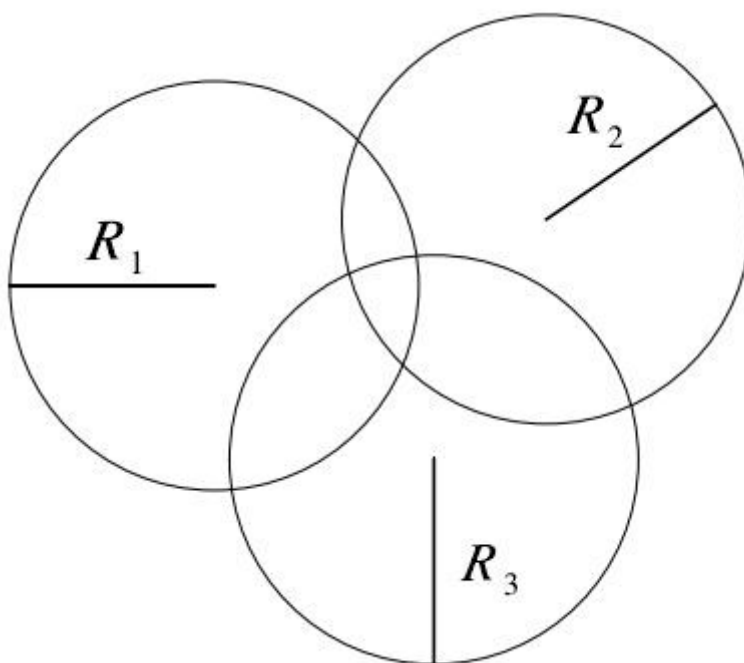
当接收到三个以上不同基站的信号时，由2.1得到终端与基站的距离之后，便可利用定位算法对基站进行定位。最广泛使用的是三边定位算法，在此基础上，改进的算法有加权三边定位算法和加权质心定位算法。

### 3.2.1 三边定位算法

在基于测距的定位算法中，三边测量法是比较简单的算法，算法原理为：平面上有三个不共线的基站 A,B,C，和一个未知终端 D，并已测出三个基站到终端D的距离分别为 $R_1$ ， $R_2$ ， $R_3$ ，则以三个基站坐标为圆心，三基站到未知终端距离为半径可以画出三个相交的圆，如图下图所示，未知节点坐标即为三圆相交点。



然而，在实际测量中，往往由于测量的误差，使三个圆并不交于一点，而相交于一块区域，如下图所示。在此种情况下，便需用其他算法进行求解，如极大似然估计法，最小二乘法进行估计，或者使用三角形质心算法。



这里，我们的算法采用最小二乘法求近似解，并针对n个基站（ $n \geq 3$ ），已知n个基站的坐标分别为  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ，未知终端坐标为  $(x, y)$ ，由以下步骤求解：

①：建立信标节点与未知节点距离方程组

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases}$$

②：上边方程组为非线性方程组，用方程组中前n-1个方程减去第n个方程后，得到线性化的方程：

$$AX = b$$

其中：

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}, b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix}$$

③：用最小二乘法求解上边方程得：

$$X = (A^T A)^{-1} A^T b$$

X 便是未知终端的坐标计算值。

### 3.2.2 加权三边定位算法

由无线信号强度渐变模型可以发现，当定位终端离基站距离越远时，接收到的RSSI值变化会越来越小，这就会导致距离越远，基站与定位终端的距离误差越大，相应的造成定位误差变大，由此，我们可以采取加权的思想，将距离小的（精确度高）赋予较大的权值，距离大的（精确度低）的赋予较小的权值。

首先将基站分组。对收集到的所有基站，经由id分为组n后，求组合数C(n, 3)，并对每组分别进行三边定位；接着根据距离越大定位误差越大的原则，赋予权值(为每个基站到定位终端测得的距离)。最后，由每个组合得到的结果加权得到最终的定位结果。

### 3.2.3 加权三角形质心定位算法

该算法的思想是对收集到的所有基站，经由id分为组n后，求组合数C(n, 3)，然后对每一个组合的三个基站，以每个基站坐标为圆心，测得的基站到定位终端距离为半径画圆，然后根据交点组成的三角形，求其质心，即为测得的终端坐标。大体如下：

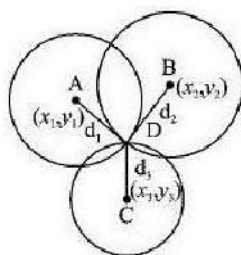


图2 存在唯一交点

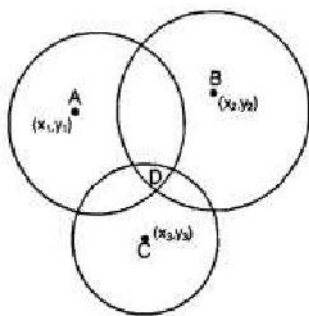


图3 两两相交1

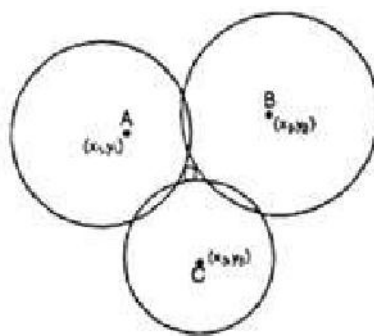


图4 两两相交2

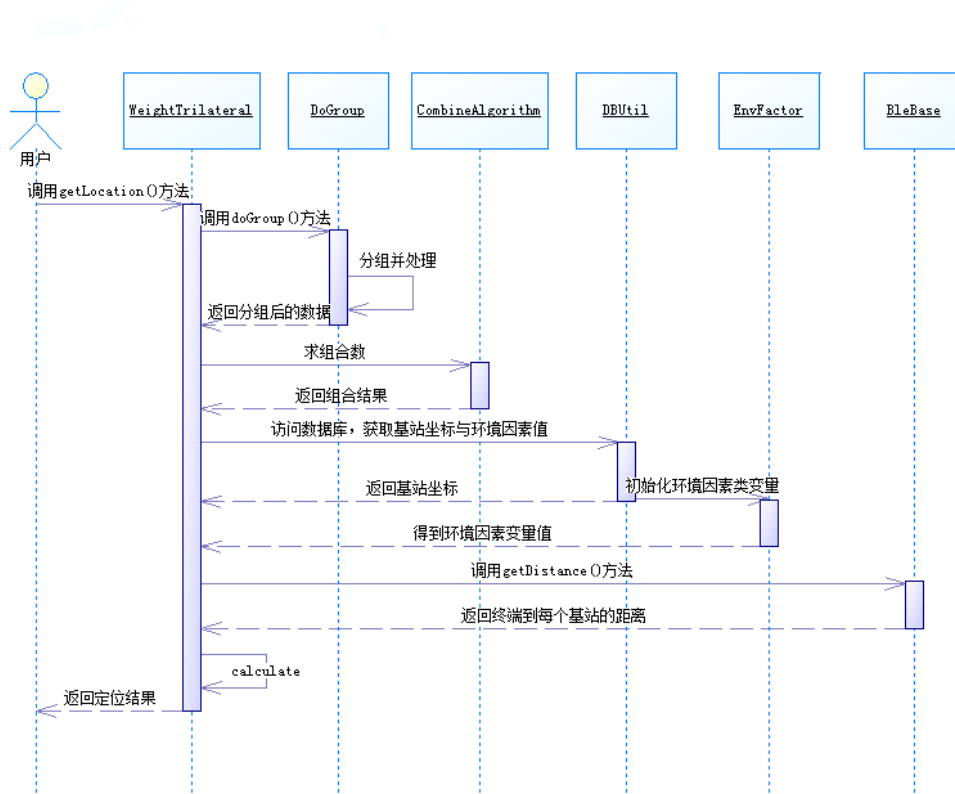
然后，根据距离越大定位误差越大的原则，赋以权值(为每个基站到定位终端测得的距离)。最后，由每个组合得到的结果加权得到最终的定位结果。

然而，由于误差的存在，且测量时位置并不在每个组合所构成的三角形中间位置，因此，当误差大时，往往所构成的圆是没有交点的（三个圆必须两两存在交点，否则不能用该方法）。

以上三个算法为了使接收到的基站RSSI更为准确，根据正态分布，去掉了RSSI值排序后两边的极端值，即对接收到的同一基站的所有RSSI值，排序，然后首尾各去掉一部分，然后求其均值，作为接收到的RSSI值测算距离。

### 3.3 程序定位算法的执行流程

定位时，使用实现了Dealer接口的相应算法对象的getLocation(String str)方法得到终端坐标，接下来以WeightTrilateral（加权三边定位算法）为例，给出其运行时序图来演示系统执行的流程：



# 二、系统运行截图

