基于蓝牙信号强度测距的室内定位方法及实现

扬州大学 物理科学与技术学院，张伟 160901137

摘要

相比室外定位技术来说，室内定位技术发展时间较晚，但由于人们对高精度室内定位的需求日益增长，成为了近年来全世界的研究热点。在复杂的室内环境中,结合最小二乘定位、位置指纹定位、设备无关定位的室内定位方法，该文提出了一种基于蓝牙低功耗（Bluetooth Low Energy，BLE）指纹-最小二乘融合的高精度室内蓝牙定位方法。采用位置估计、模型优化的方法，在 Android 上实现了室内准确定位，并将结果以二维图像形式显示出来。经实际测定，该文的定位方案在 6.51m*10.12m 的室内环境下，定位精度优于 10cm，完全可以满足日常的目标定位功能。

关键字 室内定位；蓝牙；Android；位置指纹定位；最小二乘定位

Indoor Positioning Method Based on Bluetooth Signal Strength Ranging and Its Implementation

ZHANG Wei[*] 160901137，Yangzhou University

Abstract：Compared with outdoor positioning technology, indoor positioning technology has developed late, but due to the increasing demand for high-precision indoor positioning, it has become a research hotspot in the world in recent years. In a complex indoor environment, combined with least square positioning, position fingerprint positioning, and device-independent positioning, this paper proposes a Bluetooth Low Energy (BLE) fingerprint-least-square fusion. High-precision indoor Bluetooth positioning method. Using the method of position estimation and model optimization, the indoor accurate positioning is realized on Android, and the result is displayed in the form of two-dimensional image. According to the actual measurement, the positioning scheme of this paper is better than 10cm in the indoor environment of 6.51m*10.12m, which can fully meet the daily target positioning function.

Key words：indoor positioning;Bluetooth; Android; location fingerprint location; least squares positioning

0 前言

个人穿戴设备、手机、智能手表等便携式智能设备的广泛普及，带来了随时随地获取即时个人位置及周边信息的需求，随之而来的是基于位置的服务(Location Based Services，LBS)在全球范围内的蓬勃发展。全球导航卫星系统

(GNSS) 在室外的定位应用发挥着巨大作用，但 GNSS 无线信号在空气中传播时会受到物体遮挡与阻拦[1]，尤其是存在多径效应等问题，使得 GNSS 在应用到室内定位时丧失了原本在露天环境下的优势，所以需要寻求更合适的定位技术来提高室内定位的精度。

室内定位技术[2][3]的难点集中在定位技术的多样性与定位算法的复杂性上。一般来说，定位技术与定位算法的优点不能有效结合起来，经常会出现所选用的定位技术功耗低、信号稳定，但能与之匹配的定位算法精度不高或者实现过于复杂等情况。

本文采用基于蓝牙接收信号强度指示(Received Signal Strength Indication，RSSI) 的位置指纹定位法[4]与最小二乘法相融合的方式来实现室内定位，并通过构建的定位系统验证了该融合算法，提高了定位精度。

1 定位方法

应当根据具体环境设计合理的方法实现定位，方法一般包括位置估计和位置跟踪两大类。位置估计指目标当前所在位置的估计，位置跟踪指一段时间内移动目标的运动轨迹，它在时间上具有移动性和连续性。

无论是哪种定位，都遵循以下两个基本操作流程（如图 1 所示）。本文所提出的测距定位方法中，物理测量采用蓝牙接收信号强度[5][6]的测量及数值优化实现。

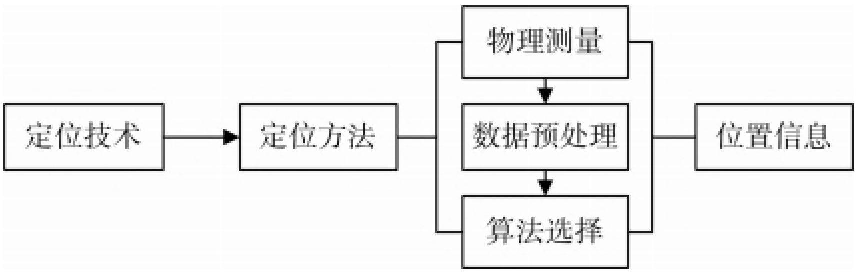
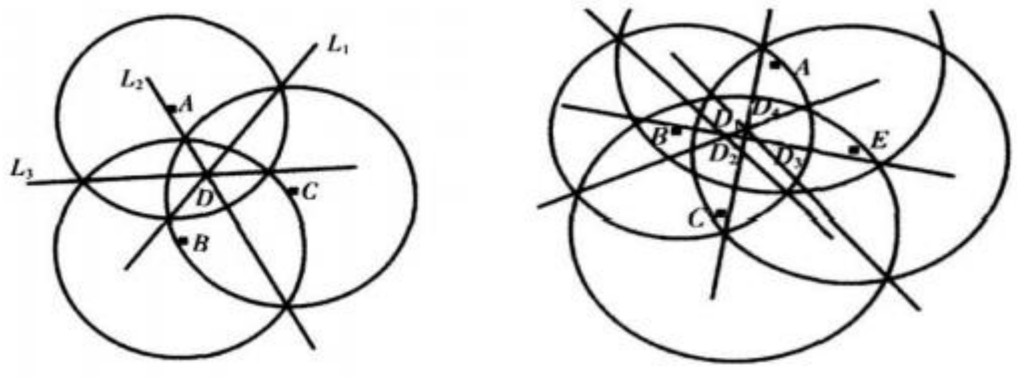


图 1 定位实现流程

1.1 四边定位法

为了进一步提高定位精度，文举等[[7]]提出四边测距定位算法，即在三边测距的基础上，再添加一个锚节点参与定位，如图 2（a）所示。根据图 2（a）的情形，每次取 3 个锚节点即 3 个圆便可对待测节点进行一次坐标估计，每组取 2 条直线估 算待测节点位置，这样，图 2（b）中便可以取 4 组圆产生待测节点 D(xD，yD)的 4 次坐标估计为 D$_1$(x$_{D1}$，y$_{D1}$)，D$_2$(x$_{D2}$，y$_{D2}$)，D$_3$(x$_{D3}$，y $_{D3}$)， D$_4$(x$_{D4}$，y$_{D4}$)，对其求质心，即可得节点 D 的坐标为：

$$\begin{cases} x_D = \dfrac{x_{D_1} + x_{D_2} + x_{D_3} + x_{D_4}}{4} \\ y_D = \dfrac{y_{D_1} + y_{D_2} + y_{D_3} + y_{D_4}}{4} \end{cases} .$$



a）三边测距定位实际情形　　　　　　　　b）四边测距定位示意图

图 2 定位方法示意图

1.2 最小二乘定位

最小二乘法是一种优化算法，是通过最小化误差的平方和来寻找数据的最佳匹配。

假设使用 n 个 Beacon，相应的位置坐标为：

$[x_1,y_1]$，$[x_2,y_2]$，…，$[x_n,y_n]$。

接收点接收到的对应的信号强度值为：（rss_1，rss_2，...，rss_n）。根据对数衰减模型得到相应的距离为（d_1，d_2，..，d_3）。

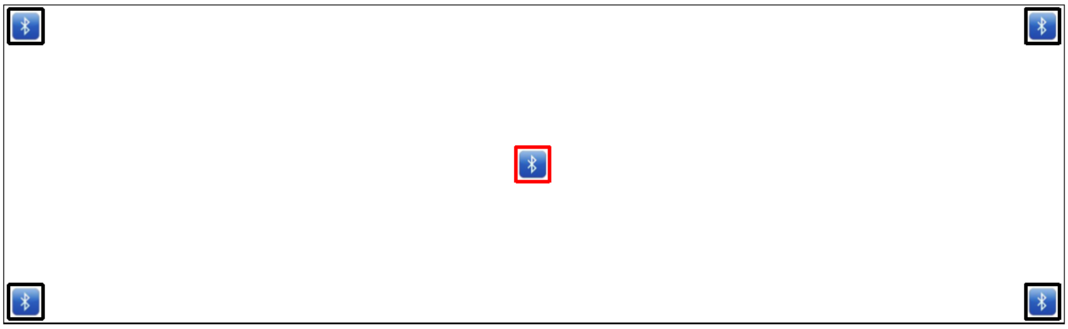


图 3 最小二乘定位（4 个参考点）

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2(x_1-x_2) & 2(y_1-y_2) \\ 2(x_1-x_3) & 2(y_1-y_3) \end{bmatrix}^{-1} \times \begin{bmatrix} (d_2^2-d_1^2)+(x_1^2-x_2^2)+(y_1^2-y_2^2) \\ (d_3^2-d_1^2)+(x_1^2-x_3^2)+(y_1^2-y_3^2) \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 2(x_1-x_2) & 2(y_1-y_2) \\ 2(x_1-x_3) & 2(y_1-y_3) \\ 2(x_1-x_4) & 2(y_1-y_4) \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} (d_2^2-d_1^2)+(x_1^2-x_2^2)+(y_1^2-y_2^2) \\ (d_3^2-d_1^2)+(x_1^2-x_3^2)+(y_1^2-y_3^2) \\ (d_4^2-d_1^2)+(x_1^2-x_4^2)+(y_1^2-y_4^2) \end{bmatrix}$$

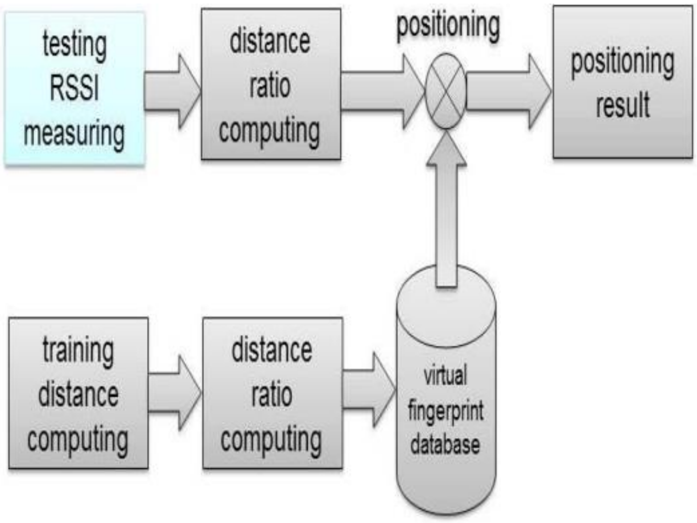
$$\Rightarrow \boldsymbol{x}_{LS} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b}$$

1.3 位置指纹定位

信号的多径传播对环境具有依赖性，呈现出非常强的特殊性。对于每个位置而言，该位置上信道的多径结构是惟一的，终端发射的无线电波经过反射和折射，产生与周围环境密切相关的特定模式的多径信号，这样的多径特征可以认为是该位置的“指纹”。基站天线阵列检测信号的幅度和相位等特性，提取多径干扰特征参数，将该参数与预先存储在数据库中的指纹数据进行匹配，找出最相似的结果来进行定位。

位置指纹定位的实施一般可以分为两个阶段：第一阶段为训练/离线阶段，主要工作是采集所需定位区域各参考节点(RP，Reference Point)位置的信号特征参数，例如信号场强、多径相角分量功率等，将一组指纹信息对应一个特定的位置形成位置指纹数据库。第二阶段为定位/在线阶段，利用接收机测定接收信号的参数，采用匹配算法来确定与数据库中哪一组数据相匹配，从而得出用户的实际位置。

基于 RSS 的技术主要有两种：传输损耗(RFPL，Radio Frequency Propagation Loss)定位法和位置指纹(LF，Location Fingerprint)定位法。前者将信号传输模型和建筑的地理信息转换成距离测量值，这样做很简单，但是受很多方面的影响. 比如距离、信号穿透墙壁时的损耗、多径传输等，所以难以建立正确的数学模型。后者则可以应付 NLOS 和多径传输的状况，但是建立和维护数据库比较麻烦。



$$\boldsymbol{F}_{vI}\left(i,j\right)=\boldsymbol{F}_{vI}\left(i\Delta x,j\Delta y\right)$$

$$=\begin{bmatrix} d_{vr}\left(1,q;i,j\right) & d_{vr}\left(2,q;i,j\right) & \cdots & d_{vr}\left(N,q;i,j\right) \end{bmatrix}$$

$$d_{vr}\left(q,q;i,j\right)=1$$

$$\boldsymbol{F}_{vII}\left(i,j\right)=\begin{bmatrix} d_{vr}\left(1,1;i,j\right) & d_{vr}\left(2,1;i,j\right) & \cdots & d_{vr}\left(N,1;i,j\right) \\ d_{vr}\left(1,2;i,j\right) & d_{vr}\left(2,2;i,j\right) & \cdots & d_{vr}\left(N,2;i,j\right) \\ \cdots & \cdots & \cdots & \cdots \\ d_{vr}\left(1,N;i,j\right) & d_{vr}\left(2,N;i,j\right) & \cdots & d_{vr}\left(N,N;i,j\right) \end{bmatrix}$$

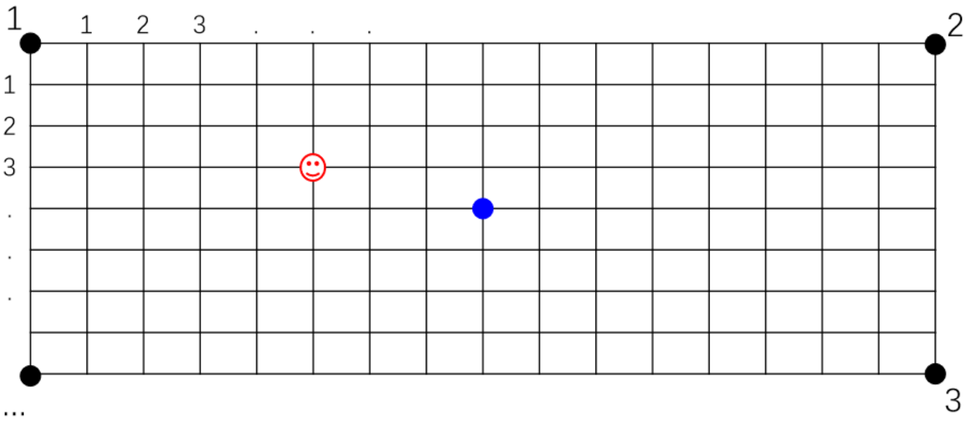
$$d_{vr}\left(p,p;i,j\right)=1,d_{vr}\left(p,q;i,j\right)=1/d_{vr}\left(q,p;i,j\right)$$

$$\boldsymbol{L}_{NN}=\begin{bmatrix} x & y \end{bmatrix}=\begin{bmatrix} i\Delta x & j\Delta y \end{bmatrix}=\underset{i,j,k}{\arg\min}\left\|\boldsymbol{F}\left(x,y\right)-\boldsymbol{F}_{v}\left(i,j\right)\right\|_2$$

1.4 设备无关定位

在前面 Android 应用程序的基础上进行。建立设备无关指纹数据库， 在空的室内平面环境均匀设计采样点，采样正方形边长为地砖边长（或 0.5 米、1 米） 。每个采样点采集三次，计算平均值，记录数据，当不携带蓝牙设备人员位于不同采样点时，采集人员用手机在固定位置采集各个参考蓝牙节点信号强度，建立训练设备无关位置指纹数据库。

然后定位目标位置，每个采样点采集一次，记录数据。使用最近邻法（或贝叶斯法）定位，计算定位误差，当不携带蓝牙设备人员位于某个待定位点时，采集人员用手机在固定位置采集 各个参考蓝牙节点信号强度，采用最近邻法（或贝叶斯法）与数据库中位置指 纹比较，确定不携带蓝牙设备人员的位置。



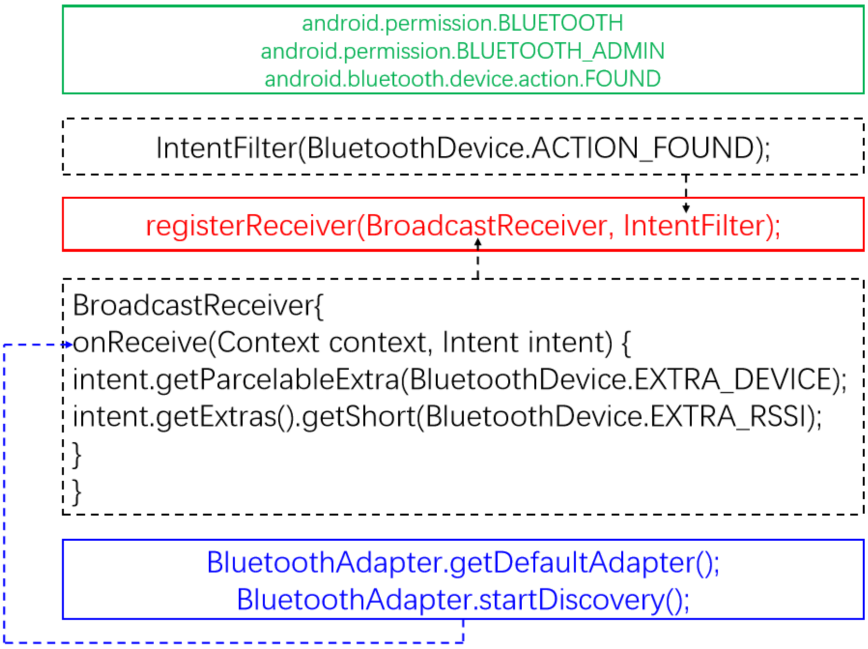
$$\boldsymbol{r} = \begin{bmatrix} r^1 & r^2 & r^3 & \cdots \end{bmatrix}$$

$$(i,j) = \underset{(i,j)}{\arg\max}\, P(i,j) = \underset{(i,j)}{\arg\max}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left|\boldsymbol{r}-\boldsymbol{r}_{ij}\right|^2}{2\sigma^2}\right)\right]$$

2 定位实现

2.1 定位实现

根据蓝牙信号强弱测距、位置估计优化的设计思路，在实际的实验机房完成 Android 程序的设计，实现 Beacon 参考点的测距（Android 程序实现如图所示）。



```
android.permission.BLUETOOTH
android.permission.BLUETOOTH_ADMIN
android.bluetooth.device.action.FOUND

IntentFilter(BluetoothDevice.ACTION_FOUND);

registerReceiver(BroadcastReceiver, IntentFilter);

BroadcastReceiver{
onReceive(Context context, Intent intent) {
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
}
}

BluetoothAdapter.getDefaultAdapter();
BluetoothAdapter.startDiscovery();
```

图 4 蓝牙测距定位程序结构示意图

首先建立室内位置指纹数据库，在室内定位空间均匀采样点根据已知位置计

算二维虚拟指纹，采样点的间隔为地砖宽度（或 0.5 米、1 米）；然后定位目标位置，在室内定位空间均匀采样点根据信号强度计算二维虚拟指纹，使用最近邻法（或贝叶斯法），与最小二乘定位法和位置指纹法比较定位误差。融合虚拟指纹法和位置指纹法及最小二乘法，取三者的平均值。

表 1 位置指纹训练数据库

F_{vII}	0.92*1	0.92*2	0.92*3	0.92*4	0.92*5	0.92*6	0.92*7	0.92*8	0.92*9	0.92*10	0.92*11
0.93*7											
0.93*6											
0.93*5											
0.93*4											
0.93*3											
0.93*2											
0.93*1											

2.2 二维图像显示

创建二维图形 Android 应用程序，将图片 bluetooth.png 复制到 AS 的“res/drawable”目录。修改前面的定位程序，将定位结果以二维图形方式显示。

手机或开发板屏幕对应整个室内定位空间，布局文件 activity_main.xml 包括 2D 图形和控件，MainActivity 访问 MyView 以传送定位点坐标 。在 activity_main 的 MyView 中增加属性 android:id="@+id/myview"，在 MainActivity 中使用 findViewById(R.id.myview)访问 MyView。

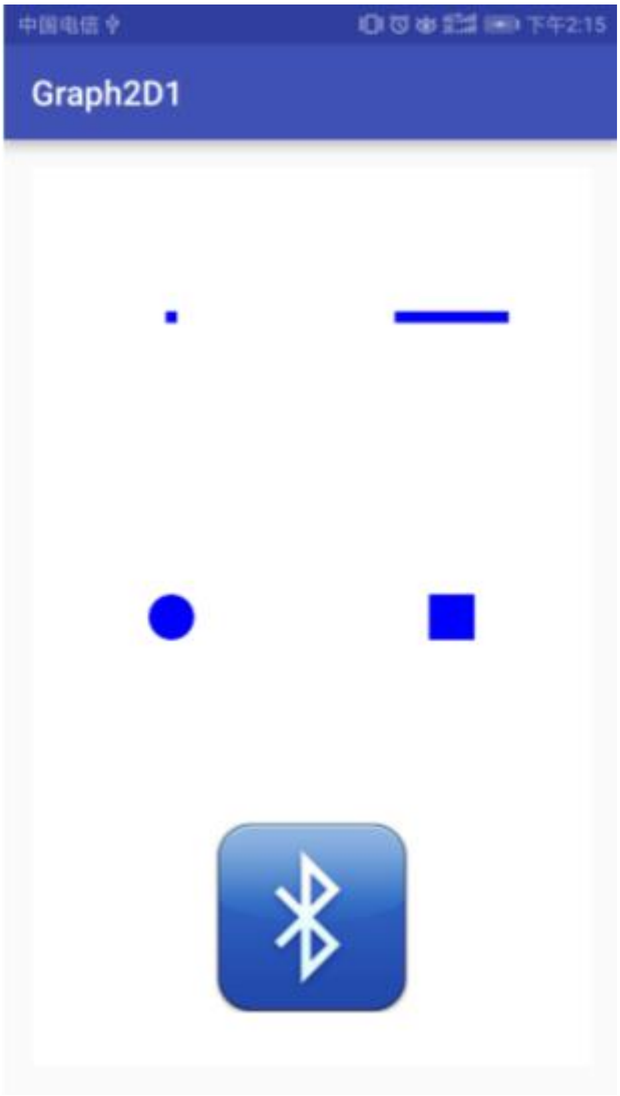


图 5 Android 二维显示界面

3 系统测试结果

实验环境为机房室内（0.93*7m × 0.92*11m），77 个 Beacon 以网格的形式均匀布置，两两间距为 0.92m，布置 7*11 个 Beacon。在中间的 1*11 个点的位置处，布置了 11 个参考点。实验中，在每个参考点测量三条数据，将数据取均值，对于每个 Beacon，利用在各个参考点的均值强度值和参考点到此 Beacon 的距离，来训练对数衰减模型。

本文从测试区域中，选取了 11 个测试点，分别使用三边测量法和提出的最小二乘法计算定位结果。训练误差表如 表 1 所示，从该表可以看出最小二乘拟合定位的平均定位误差和位置指纹法的平均误差相差不大，但它能够有效地降低最大定位误差，使得定位更加稳定，误差波动更小。

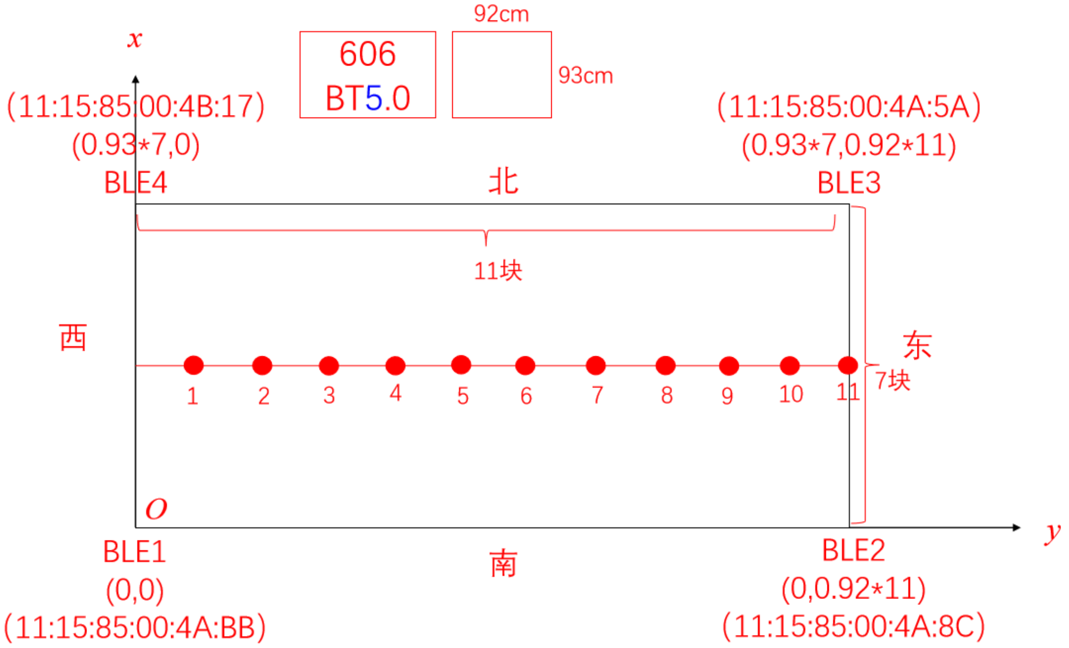


图 6 测试环境示意图

表 2 三种定位方法测试对比

d(m)距离		四边定位法		最小二乘定位		位置指纹定位		设备无关定位		WiFi 定位	
x	y	结果（x’，y’）	误差	结果（x’，y’）	误差	结果（x’，y’）	误差	结果（x’，y’）	误差	结果（x’，y’）	误差
0.93*3.5	0.92*1										
0.93*3.5	0.92*2										
0.93*3.5	0.92*3										
0.93*3.5	0.92*4										
0.93*3.5	0.92*5										
0.93*3.5	0.92*6										
0.93*3.5	0.92*7										
0.93*3.5	0.92*8										
0.93*3.5	0.92*9										
0.93*3.5	0.92*10										
0.93*3.5	0.92*11										

定位结果的误差累积分布图如图 3 所示，从图 3 可以看出，在大部分情况下，本文所采用的最小二乘法的误差均小于位置指纹定位法，说明了本文方法的有效性。

$$f\left(e;\mu,\sigma\right)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{\left(e-\mu\right)^{2}}{2\sigma^{2}}\right)$$

$$e=\sqrt{\left(x-x_{t}\right)^{2}+\left(y-y_{t}\right)^{2}}$$

$$F\left(e;\mu,\sigma\right)=\int_{-\infty}^{e}f\left(e;\mu,\sigma\right)de$$

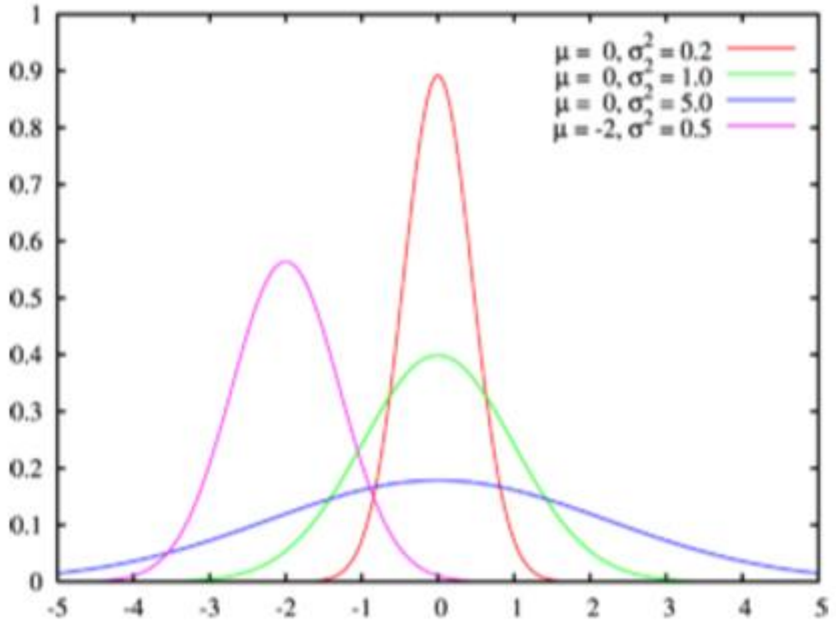


图 7 误差累计概率

4 结束语

本文提出的室内测距定位方法，通过位置估计、模型优化，在 Android 上实现了室内准确定位，并将结果以二维图像形式显示出来。经实际测定，该方法在 6.51m*10.12m 的室内环境下，定位精度优于 10cm，完全可以满足日常的目标定位功能。

非常感谢老师和同学在本方法实现的过程中给予了非常大的帮助，没有他们的支持，该方法很难实现。另外，由于作者水平所限，文中难免有不足之处，欢迎各位读者反馈宝贵建议和意见。

参考文献

[1] 郝雨时,徐爱功,隋心 . GNSS / UWB 高精度室内外组合定位方法[J]. 导航定位学报,2016,4(4):17-23.Hao Yushi, Xu Aigong, Sui Xin. Method of indoor and outdoor integrated precise positioning based on GNSS /UWB[J]. Journal of Navigation and Positioning,2016,4(4):17-23.

[2] 王星星,丛思安.室内定位研究方法综述[J/OL].软件导刊:1-4[2019-06-05].http://kns.cnki.net/kcms/detail/42.1671.TP.20190527.1545.068.html.

[3] 张胜利,焦诚,付永恒.室内定位方法和技术综述[J].数字技术与应用,2018,36(10):73-74+76.

[4] 吴雅琴,杨硕,师兰兰.基于位置指纹与 PDR 融合的室内定位算法研究[J/OL].矿业科学学报,2019(05):1-7[2019-06-05].https://doi.org/10.19606/j.cnki.jmst.2019.05.010.

[5] 李明阳.基于蓝牙的室内定位技术研究[J].科技经济导刊,2019,27(08):27.

[6] 杨保,张鹏飞,李军杰,路朋.一种基于蓝牙的室内定位导航技术[J/OL].测绘科学:1-12[2019-06-05].http://kns.cnki.net/kcms/detail/11.4415.P.20190409.1940.061.html.

[7] 文举,金建勋,袁海.一种无线传感器网络四边测距定位算法[J].传感器与微系统,2008(05):108-110+113.

[8] 王旭强,万亚平,李洪飞,耿家兴.基于 WiFi 指纹的室内定位系统[J].电脑编程技巧与维护,2019(05):133-135.

附件 核心程序

1. MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private static final double classroom_y_max=0.92*11;
    private static final double classroom_x_max=0.93*7;
    private static final String address0="98:D3:31:08:26:C8";
    private static final String address1="80:B3:92:D2:BA:97";
    private static final String address2="80:B3:92:CE:DF:41";
    private static final String address3="80:B3:92:CE:B0:95";
    private static final String address4="80:B3:92:D2:E1:2C";
    private static final String filename="RssiRecord.db";
    private String filePath;
    TextView tv;
    SQLiteDatabase DB;
    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        getMenuInflater().inflate(R.menu.menu_main,menu);
        return  super.onCreateOptionsMenu(menu);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        // Register the BroadcastReceiver
        IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
        registerReceiver(mReceiver,filter);

        tv = (TextView) findViewById(R.id.text_info);
        BluetoothAdapter mBluetoothAdapter =BluetoothAdapter.getDefaultAdapter();
        mBluetoothAdapter.startDiscovery();

        filePath=getFilesDir()+"/" + filename;//
        Thread thread=new Thread(runnable);
        thread.start();

    }

    public void printTest (String s) {
        tv.append(s);
    }
    Runnable runnable=new Runnable()
    {
        @Override
        public void run() {
            while(true){
                BluetoothAdapter mBluetoothAdapter
=BluetoothAdapter.getDefaultAdapter();

                if(!mBluetoothAdapter.isDiscovering())
                   mBluetoothAdapter.startDiscovery();

                try{
                    Thread.sleep(400);
                }
                catch (InterruptedException e)
                {
                    tv.setText("sleep failed!");
                }
            }
        }
    };

    long fond_bluetooth1_times=0;
    long fond_bluetooth2_times=0;
    long fond_bluetooth3_times=0;
    long fond_bluetooth4_times=0;
    boolean fFund_bluetooth1=false;
    boolean fFund_bluetooth2=false;
    boolean fFund_bluetooth3=false;
    boolean fFund_bluetooth4=false;
    short []Rssi={0,0,0,0};
    long All_Rssi=0,Diff_Rssi=500;
    String Similar_position;
    private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
        public void onReceive(Context context, Intent intent) {
            // When discovery finds a device
            if (BluetoothDevice.ACTION_FOUND.equals(intent.getAction())) {
                // Get the BluetoothDevice object from the Intent
                BluetoothDevice device
=intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                // Add the name and address to an array adapter to show in a ListView
                 BluetoothAdapter mBluetoothAdapter
```

```
=BluetoothAdapter.getDefaultAdapter();

                if((device.getAddress()).equals(address0)) {
                    //fFund_bluetooth1=true;
                    fond_bluetooth1_times++;
                    Rssi[0]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);

                    printTest("蓝牙1发现次数:"+fond_bluetooth2_times+" 强度:" +
Rssi[0] + "\n");
                    Rssi[0]*=-1;
                    All_Rssi=Rssi[0];

fFund_bluetooth1=fFund_bluetooth2=fFund_bluetooth3=fFund_bluetooth4=true;

                }
                else if((device.getAddress()).equals(address2)) {

                    fond_bluetooth2_times++;
                    fFund_bluetooth2=true;
                    Rssi[1]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
                    //printTest("蓝牙2发现次数:"+fond_bluetooth2_times+" 强度:" +
Rssi[1] + "\n");
                    Rssi[1]*=-1;
                }
                else if((device.getAddress()).equals(address3)) {

                    fond_bluetooth3_times++;
                    fFund_bluetooth3=true;
                    TextView tv = (TextView) findViewById(R.id.text_info);
                    tv.setText("");
                    Rssi[2]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
                    //printTest("蓝牙3发现次数:"+fond_bluetooth2_times+" 强度:" +
Rssi[2] + "\n");
                    Rssi[2]*=-1;
                }
                else if((device.getAddress()).equals(address4)) {
                    fond_bluetooth4_times++;
                    fFund_bluetooth4=true;
                    Rssi[3]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
                    //printTest("蓝牙3发现次数:"+fond_bluetooth2_times+" 强度:" +
Rssi[3] + "\n");
                    Rssi[3]*=-1;
                }
```

```
if(fFund_bluetooth1&&fFund_bluetooth2&&fFund_bluetooth3&&fFund_bluetooth4)
                {
                    mBluetoothAdapter.cancelDiscovery();
                    TextView tv = (TextView) findViewById(R.id.text_info);
                    tv.setText("");
                    fFund_bluetooth1=false;
                    fFund_bluetooth2=false;
                    fFund_bluetooth3=false;
                    fFund_bluetooth4=false;
                    //All_Rssi=Rssi[0]+2*Rssi[1]+3*Rssi[2]+4*Rssi[3];

                    File file=new File(filePath);
                    if(!file.exists())
                    {
                        printTest("数据库不存在！\n请在采样活动里记录数据后重试！");
                    }
                    else
                    {
                        DB = SQLiteDatabase.openDatabase(getFilesDir() +
"/RssiRecord.db", null, SQLiteDatabase.OPEN_READONLY);
                        Cursor cursor= DB.rawQuery("select * from RssiPosition",null);
                        if (cursor.moveToFirst() == false)
                        {
                            printTest( "您的表中无数据！！！" + "\n");
                        }
                        else
                        {

for(cursor.moveToFirst();!cursor.isAfterLast();cursor.moveToNext())
                            {
                                String position =
cursor.getString(cursor.getColumnIndex("Position"));
String rssi = cursor.getString(cursor.getColumnIndex("Rssi"));
long long_rssi=Integer.valueOf(rssi).intValue();
if(Math.abs(long_rssi-All_Rssi)<=Diff_Rssi)
{
     Diff_Rssi=Math.abs(long_rssi-All_Rssi);
     Similar_position=position;
     }
}
printTest("定位坐标：" + Similar_position + "\n特征差值：" + Diff_Rssi + "\n");
                            String []position=Similar_position.split("\\s+");
                            MyView myview=(MyView)findViewById(R.id.Myview);
                            myview.x_per=Float.parseFloat(position[0]);
                            myview.y_per=Float.parseFloat(position[1]);
                            myview.postInvalidate();
                            Diff_Rssi=500;
                        }
                    }
                }
            }
        }
    };
@Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.Record_Rssi:
                //Toast.makeText(this, "You clicked study", Toast.LENGTH_SHORT).show();
                Intent intent=new Intent(this,RssiStudyActivity.class);
                startActivity(intent);
                break;
            case R.id.use_help:
                Intent intent1=new Intent(this,HelpActivity.class);
                startActivity(intent1);
                break;
            default:
        }
        return true;
    }
}
```

2. RssiStudyActivity.java

```
public class RssiStudyActivity extends AppCompatActivity implements
View.OnTouchListener{
    public double current_x_per=0;
    public double current_y_per=0;
    private static final String selectSql = "select _id, Position, Rssi from
RssiPosition";
    private static final String insertSql = "insert into RssiPosition(Position, Rssi)
values(?,?)";
    private static final String filename="RssiRecord.db";
    private RelativeLayout Rl_touch;
    String filePath;
    public MyView myview;
    private TextView tv;
    SQLiteDatabase DB;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rssi_study);
        //Toast.makeText(this, "You clicked study first", Toast.LENGTH_SHORT).show();
        IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
        registerReceiver(mReceiver,filter);

        Rl_touch=(RelativeLayout)findViewById(R.id.Rl_touch);//
        Rl_touch.setOnTouchListener(this);

        BluetoothAdapter mBluetoothAdapter =BluetoothAdapter.getDefaultAdapter();
        mBluetoothAdapter.startDiscovery();

        myview=(MyView)findViewById(R.id.Myview);
        tv = (TextView) findViewById(R.id.text_info);

        filePath=getFilesDir()+"/" + filename;//
        File file=new File(filePath);
        if(!file.exists())
        {
            DB = SQLiteDatabase.openDatabase(filePath, null,
SQLiteDatabase.CREATE_IF_NECESSARY);
            String createSql = "create table RssiPosition(_id integer primary key
autoincrement, Position, Rssi)";
            DB.execSQL(createSql);
            Toast.makeText(this, "数据库创建成功！", Toast.LENGTH_SHORT).show();
        }
        else
        {
            DB = SQLiteDatabase.openDatabase(getFilesDir()+"/RssiRecord.db", null,
SQLiteDatabase.CREATE_IF_NECESSARY);
            Toast.makeText(this, "数据库已存在！", Toast.LENGTH_SHORT).show();
        }

        Thread thread=new Thread(runnable);
        thread.start();

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)//
    {
        getMenuInflater().inflate(R.menu.menu_sql,menu);
        return  super.onCreateOptionsMenu(menu);
    }

    public void printTest (String s) {
        tv.append(s);
    }

    Runnable runnable=new Runnable() {
        @Override
        public void run() {
            while(true){
                try
                {    Thread.sleep(500); }
                catch (InterruptedException e)
                { tv.setText("sleep failed!"); }
            }
        }
    };

    long fond_bluetooth1_times=0;
    long fond_bluetooth2_times=0;
    long fond_bluetooth3_times=0;
    long fond_bluetooth4_times=0;
    boolean fFund_bluetooth1=false;
    boolean fFund_bluetooth2=false;
    boolean fFund_bluetooth3=false;
    boolean fFund_bluetooth4=false;
    boolean Recording=false;
    String first_id;//
    String last_id;//
    boolean finish=false;
    short [] Rssi={0,0,0,0};
    long All_Rssi=0;
    private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
        public void onReceive(Context context, Intent intent) {
            // When discovery finds a device
            if (BluetoothDevice.ACTION_FOUND.equals(intent.getAction())) {
                // Get the BluetoothDevice object from the Intent
                BluetoothDevice device
=intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                // Add the name and address to an array adapter to show in a ListView
                BluetoothAdapter mBluetoothAdapter
=BluetoothAdapter.getDefaultAdapter();
                if((device.getAddress()).equals(address0)) {
                    //fFund_bluetooth1=true;
                    fond_bluetooth1_times++;
                    Rssi[0]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
                    printTest("蓝牙 1 发现次数:"+fond_bluetooth1_times+" 强度:" +
Rssi[0] + "\n");
                    Rssi[0]*=-1;
                    All_Rssi+=Rssi[0];
                    finish=true;
                }
                else if((device.getAddress()).equals(address2)) {

                    fond_bluetooth2_times++;
                    fFund_bluetooth2=true;
                    Rssi[1]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
                    //printTest("蓝牙 2 发现次数:"+fond_bluetooth2_times+" 强度:" +
Rssi[1] + "\n");
                    Rssi[1]*=-1;
                }
                else if((device.getAddress()).equals(address3)) {

                    fond_bluetooth3_times++;
                    fFund_bluetooth3=true;
                    Rssi[2]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
                    //printTest("蓝牙 3 发现次数:"+fond_bluetooth3_times+" 强度:" +
Rssi[2] + "\n");
                    Rssi[2]*=-1;
                }
                else if((device.getAddress()).equals(address4)) {

                    fond_bluetooth4_times++;
                    fFund_bluetooth4=true;
                    Rssi[3]=intent.getExtras().getShort(BluetoothDevice.EXTRA_RSSI);
                    //printTest("蓝牙 4 发现次数:"+fond_bluetooth4_times+" 强度:" +
Rssi[3] + "\n");
                    Rssi[3]*=-1;
                }

if(fFund_bluetooth1&&fFund_bluetooth2&&fFund_bluetooth3&&fFund_bluetooth4)
                {
                    //mBluetoothAdapter.cancelDiscovery();
                    //tv.setText("");
                    fFund_bluetooth1=false;
                    fFund_bluetooth2=false;
                    fFund_bluetooth3=false;
                    fFund_bluetooth4=false;
                    All_Rssi+=Rssi[0]+2*Rssi[1]+3*Rssi[2]+4*Rssi[3];
                    //save Rssi and position
                    finish=true;

                }
            }
        }
    };
public void RecordRssi(View view) {
    tv.setText("");
    Button button=(Button) findViewById(R.id.RecordRssi);
    Recording=!Recording;
    if(Recording) {
        button.setText("停止记录");
        finish=false;
        fond_bluetooth1_times=fond_bluetooth2_times=0;
        fond_bluetooth3_times=fond_bluetooth4_times=0;
        fFund_bluetooth1=false;
        fFund_bluetooth2=false;
        fFund_bluetooth3=false;
        fFund_bluetooth4=false;
        All_Rssi=0;

    }
    else{
        button.setText("开始记录");
        if(finish){
            finish=false;
            InsertSQL(current_x_per,current_y_per,All_Rssi/fond_bluetooth1_times);
            fond_bluetooth1_times=fond_bluetooth2_times=0;
            fond_bluetooth3_times=fond_bluetooth4_times=0;
            Toast.makeText(this, "本次选中的点已记录完成！",
Toast.LENGTH_SHORT).show();
        }
    }
@Override
    public  boolean onTouch(View v, MotionEvent event){
        tv.setText("");
        current_x_per=event.getX()/v.getWidth();
        current_y_per=event.getY()/v.getHeight();
        myview.x_per=(float)current_x_per;
        myview.y_per=(float)current_y_per;
        printTest("\nx_per:"+ String.format("%.2f",current_x_per*100) +"%\n");
        printTest("y_per:"+ String.format("%.2f",current_y_per*100) +"%\n");
        myview.postInvalidate();
        return true;

    }

    public void InsertSQL(double x,double y,long Rssi)
    {
         String position=String.format("%.2f",x)+" "+String.format("%.2f",y);
        String rssi=String.valueOf(Rssi);
        DB.execSQL(insertSql, new  String[]{position, rssi});

    }

    public void UpdataSQL(String str_id,double x,double y,long Rssi)
    {
        String updateSql = "update RssiPosition set Position=?,Rssi=? where _id=?";
        String position=String.format("%.2f",x)+" "+String.format("%.2f",y);
        String rssi=String.valueOf(Rssi);
        DB.execSQL(updateSql, new String[]{position,rssi, str_id});
    }

    public void DeleteSQL(String str_id,String position,String rssi){
        if(str_id!=null) {
            String deleteSql = "delete from RssiPosition where _id=?";
            DB.execSQL(deleteSql, new String[]{str_id});
            //DB.delete(RssiPositon,String whereClause, String[] whereArgs);
        }
        if(position!=null){
            String deleteSql = "delete from RssiPosition where Position=?";
            DB.execSQL(deleteSql, new String[]{position});
        }
        if(rssi!=null){
            String deleteSql = "delete from RssiPosition where rssi=?";
            DB.execSQL(deleteSql, new String[]{rssi});
        }
    }
    public boolean PrintSQL(){
        ArrayList<HashMap<String,String>> AL= new ArrayList<HashMap<String,String>>();
        try {
            Cursor cursor = DB.rawQuery("select * from RssiPosition", null);
            if (cursor.moveToFirst() == false)
            {
                Toast.makeText(this, "您的表中无数据！！！",
Toast.LENGTH_SHORT).show();
            }
            else
            {
                for (cursor.moveToFirst(); !cursor.isAfterLast(); cursor.moveToNext())
{
                    HashMap<String, String> HM = new HashMap<String, String>();
                    String position =
cursor.getString(cursor.getColumnIndex("Position"));
                    String rssi = cursor.getString(cursor.getColumnIndex("Rssi"));
                    String str_id = cursor.getString(cursor.getColumnIndex("_id"));
                    HM.put("Position", position);
                    HM.put("Rssi", rssi);
                    AL.add(HM);
                    printTest(str_id + " " + position + "  " + rssi + "\n");
                }
            }
        }
        catch (Exception e){
            Toast.makeText(this, "您的表可能为空！！！", Toast.LENGTH_SHORT).show();
        }
        return  true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        if(item.getItemId()==R.id.clear_print)
            tv.setText("");
        Cursor cursor=DB.rawQuery("select * from RssiPosition",null);

        if (cursor.moveToFirst() == false)
        {
            Toast.makeText(this, "您的表中无数据！！！", Toast.LENGTH_SHORT).show();
            return false;
        }
        else
        {
            cursor.moveToFirst();
            first_id=cursor.getString(cursor.getColumnIndex("_id"));
            cursor.moveToLast();
            last_id=cursor.getString(cursor.getColumnIndex("_id"));
        }

        switch (item.getItemId()) {
            case R.id.printSQL:
                PrintSQL();
                break;
            case R.id.delete_first:
                DeleteSQL(first_id,null,null);
                break;
            case R.id.updata_last:
                Button button=(Button) findViewById(R.id.RecordRssi);
                if(Recording)
                {
                    Recording=false;
                    button.setText("开始记录");
                    if(finish){
                        finish=false;

UpdataSQL(last_id,current_x_per,current_y_per,All_Rssi/fond_bluetooth1_times);
                        fond_bluetooth1_times=fond_bluetooth2_times=0;
                        fond_bluetooth3_times=fond_bluetooth4_times=0;
                        Toast.makeText(this, "本次选中的点已更新完成！",
Toast.LENGTH_SHORT).show();
                    }
                }
                break;
            default:
        }
        return true;
    }
}
```