

# 面向现实的移动 SNS 系统<sup>①</sup>

唐磊<sup>1</sup>, 陈志刚<sup>1,2</sup>, 曾锋<sup>1</sup>, 郭宏杰<sup>1</sup>, 石文海<sup>1</sup>, 宋昂<sup>1</sup>, 刘鑫<sup>1</sup>

<sup>1</sup>(中南大学 软件学院, 长沙 410075)

<sup>2</sup>(中南大学 信息工程学院, 长沙 410083)

**摘要:** 蓝牙技术、GPS 定位技术, 重力加速器技术的不断发展为移动社交网络服务的产生和发展提供了新的契机。以面向现实为应用需求, 探讨了 MSNS (Mobile Social Network Services) 的应用开发技术, 设计了系统的体系结构, 基于 Android 操作系统研究了其中关键功能的设计与实现。面向现实移动 SNS 系统拉近了人与人之间的距离, 使 SNS 从“虚拟”走向“现实”。

**关键词:** 移动 SNS; 面向现实; android; 传感器

## Reality-Oriented Mobile SNS System

TANG Lei<sup>1</sup>, CHEN Zhi-Gang<sup>1,2</sup>, ZENG Feng<sup>1</sup>, GUO Hong-Jie<sup>1</sup>, SHI Wen-Hai<sup>1</sup>, SONG Ang<sup>1</sup>, LIU Xin<sup>1</sup>

<sup>1</sup>(School of Software, Central South University, Changsha 410075, China)

<sup>2</sup>(College of Information Science and Engineering, Central South University, Changsha 410083, China)

**Abstract:** With the development of the wireless technology, such as Bluetooth, GPS and gravitational acceleration sensing, there is a new opportunity for the development of the MSNS(Mobile Social Network Services). After discussing the application development of mobile social network services based on the background of reality world, this paper introduces the architecture and design of MSNS. In the system, the key function is developed based on android operation system. The MSNS system of reality oriented cuts the distance between people, making people go out from virtual social network into realistic social network.

**Key words:** mobile social network service(MSNS); reality-oriented; android; sensor

## 1 引言

SNS(全称 Social Networking Services, 即社会性网络服务)已经成为影响全球用户的新的生活方式, 形成了粘性很高的网络社区。随着 3G 应用的快速推进和手机终端性能的提升, 手机逐渐成为个人信息处理中心, 互联网从 PC 到手机的趋势正在加速。广大的用户群体及市场需求, 促使了手机互联网的迅猛发展。从而, 移动 SNS 应运而生, 并成为当前的关注热点。

面向现实的移动 SNS 系统, 除了具有诸如个人主页、留言、日志等传统的 SNS 功能外, 还可以通过使用手机蓝牙, GPS 更新位置感应周围用户, 通过手机重力加速器、方向等传感器, 摄像头进行相互感应并交换实时数据, 搭建起能辅助周围人进行实时互动、

沟通的移动 SNS 平台。希望将现实生活当中人与人之间的交互进一步延伸, 帮助移动用户建立更紧密和快捷的联系。与传统的面向互联网的 SNS 相比, 面向现实的移动 SNS 充分利用了 LBS(Location-Based Service)即基于位置的服务, 使用户可以在任何时间、任何地点获得基于定位信息的位置服务<sup>[1]</sup>, 极大增强了用户的真实性、地域性、和交互的实时性<sup>[2]</sup>。

## 2 系统总体架构设计

面向现实移动 SNS 采用 Servlet+Android+Mysql 的架构模式(如图 1 所示)。Android 是基于 Linux 平台的开源手机操作系统的名称, 自顶向下分为 4 个层次, 即应用层、应用框架层、组件库层、虚拟机和 Linux

① 基金项目:国家自然科学基金(60873082);国家大学生创新性实验计划(LA10006)

收稿时间:2011-05-18;收到修改稿时间:2011-07-04

内核层。Android SDK 给开发者提供了必要的工具和 API,方便用户编程,程序员可以很容易的针对硬件底层做开发<sup>[3]</sup>。Tomcat web 服务器支持最新的 servlet 和 JSP 规范,技术先进、性能稳定,而且开源免费,是目前比较流行的 Web 应用服务器。Mysql 数据库是一个小型关系型数据库管理系统,它体积小、速度快,开源免费,适合于中小型网站应用。

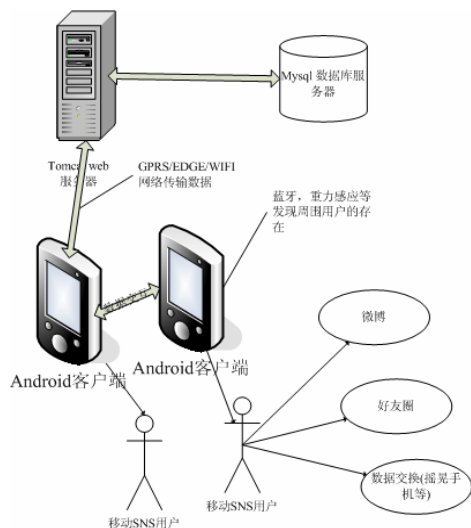


图 1 系统架构图

面向现实的移动 SNS 主要实现用户之间方便快捷真实交流的功能,其最主要特征就是基于手机传感器并且是基于现实生活,通过不同的移动设备之间进行匹配,建立连接以实现广播个人数据,和其他设备同步信息的功能,从下面三个方面进行 SNS 活动。

**社交广播:**在现实的周围人群中广播公开数据。移动终端之间“一对多”的通讯模式,即其中每一个终端向外广播数据,周围的终端无须验证就可以接收到该终端发出的信息;

**社交单播:**在现实中与人交换私有数据。移动终端之间“一对一”的通讯模式,即终端有选择性的向另一个终端发送数据。

**机会网络<sup>[4]</sup>传播:**在移动终端之间通过蓝牙、WiFi 建立起机会网络,通过中间节点以“存储-携带-转发”或者“存储-转发”的路由模式实现终端节点之间的通信。

具体而言,本系统将有如下功能:

(1) 面向现实的移动 SNS 具备普通 SNS 所具有的功能,例如微博,群组等。

(2) 通过蓝牙,能够搜索到用户附近的在该社区的其他用户,进而获取用户信息或者进行蓝牙聊天等;

(3) 通过碰撞(传感器应用),可以进行用户匹配,成功匹配后可以获取社区用户信息或者交换用户自己期望交换的数据信息;

(4) 通过 GPS、WiFi 定位服务来确定当前的用户位置坐标,通过 GPRS/EDGE/WIFI 网络从 Google Maps 获取当前位置及周围一定范围内的其他用户的地理位置信息,并将信息一起转化为固定格式以列表或者地图形式显示在用户的手机屏幕上;

(5) 通过摄像头摄取含有用户唯一标识的二维码名片可以获取到相关用户信息和社区信息,进而进行数据信息的交互。

### 3 核心功能的设计与实现

#### 3.1 技术路线

手机产生的重力加速数据的匹配算法,采用前期对数据的统计,过滤噪音,进行特征提取,降噪处理,建立模型,进行拟合。对于空间时间的重合感应数据处理上,因为网络信号问题导致的延时等噪音(例如在同一个时间段内有多方同时发送数据到服务器,导致难以确定正在通讯的目标双方),所以服务器必须对其筛选和过滤,也可以利用 GPS 辅助定位确定正在连接的双方。

移动网络的建立上,将建立起基于社区的机会网络机制存储和转发消息数据<sup>[5]</sup>,文章[2]介绍了机会网络的组网方式以及常见的几种机会网络的信息传送方式,文章[3]介绍了一种专门基于社区机会网络的消息传输算法。

定位服务上,借助全球定位系统及基站定位机制进行过滤噪音,采用 android 平台下 GPS 硬件和 Google 提供的基站定位辅助系统服务。

对于二维码,生成符合快速响应矩阵码 QR Code 标准<sup>[6]</sup>的方式有通过 GoogleApi 的方式和移动设备本地生成的方式,本系统中采用的是用 SwetakeQRCode<sup>[7]</sup>第三方函数库本地生成的方式来实现,而对于扫描二维条形码进行解码的方式则采用了 ZXing<sup>[8]</sup>开源项目(ZXing 是一个开源 Java 类库用于解析多种格式的 1D/2D 条形码。目标是能够对 QR 编码、Data Matrix、UPC 的 1D 条形码进行解码。其提供了多种平台下的客户端包括: J2ME、J2SE 和 Android)。

## 3.2 基础知识

### 3.3.1 传感器应用开发

接近硬件的传感器方面上, 由于 Android SDK 中提供的面向硬件的特性支持, `android.hardware.SensorManager` 类允许了自由访问 Android 平台的传感器, 以及 `android.hardware.SensorListener` 接口能监视硬件中的传感器的变化, 是传感器应用的中心, 它包括两个必需方法: `onSensorChanged(int sensor, float values[])`, 传感器值更改时被调用, 在这个方法实现中, 可以获取传感器数据的变化进而进行相应的处理; 当传感器的准确性更改时, 将调用 `onAccuracyChanged(int sensor, int accuracy)` 方法<sup>[9]</sup>。

首先, 我们要了解 Android `SensorEvent` API 中的坐标系统, 它与手机默认屏幕的方向相关, 坐标不随设备的屏幕变化而变化。如图 2 所示<sup>[10]</sup>:

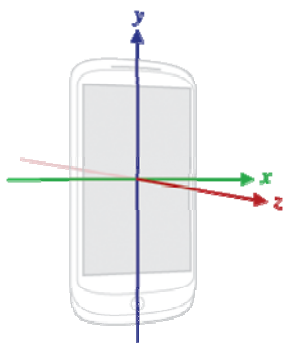


图 2 Android 平台坐标系统

对于传感器的处理主要是两个方面。其一为降噪处理。日常生活当中, 人们使用手机时一般是拿在手上产生的噪音更多。为了使该算法尽可能的适用于不同的应用环境, 需要对大部分的使用环境进行测试。首先, 由 4 个人 (8 个或更多也可以) 各自对不同手机使用环境进行数据收集, 因为人们在同样的场合使用手机的习惯是不一样的。其次, 对采集来的这些数据分别进行特征提取。最后, 使用提取的特征模型进行识别检验。其二为感应器的敏感度设置, 在 Android 中提供了四种延迟级别中, 要考虑到节省电能和逻辑处理, 根据实验的效果看来, `SENSOR_DELAY_GAME` 延迟比较适合手机收集重力感应数据进行多个手机匹配算法的精度要求。

(1) 加速度传感器, 用来测量当物体在加速过程中作用在物体上在相对设备的  $x$ ,  $y$ ,  $z$  三个坐标轴上的

力。从中可以得到的加速度传感器 3 个方向上的值: `accelerometerX`:  $x$  轴方向的加速度-重力加速度在  $x$  轴上的分量。`accelerometerY`:  $y$  轴方向的加速度-重力加速度在  $y$  轴上的分量。`accelerometerZ`:  $z$  轴方向的加速度-重力加速度在  $z$  轴上的分量。但是, 下列两点值得注意:

① 当移动设备处于静止状态时, 加速度计能够计算它相对于地平面的朝向。设备处于水平完全静止或者匀速运动, 这时的它会在垂直方向的轴上输出一个值, 那就是重力加速度, 接近重力加速度  $g$ , 如果设备没有处于绝对水平, 静止或者匀速运动状态时, 此时  $x, y, z$  轴的数值为自身运动的线性加速度和重力加速度的分量的和。此时可以通过求分量于三个轴的角度来得知手机旋转的角度。

② 如果设备沿着任意方向加速时 (或者纯水平旋转, 或者匀速水平变向运动), 而且不知道设备当前的方向角度的话, 无法进行分解重力, 无法确定设备的线性加速度部分。因为没办法知道设备目前的方向角度信息, 无法进行重力加速度的分量分解。这种情况下需要方向传感器的辅助。

(2) 方向传感器, 是测量手机相对地面的方向, 同样是  $x, y, z$  3 个轴的数值。从方向传感器可以获取下列 3 个数据: `orientation0`:  $y$  轴在水平面上的投影偏离正北方向的角度, 范围  $0 \sim 359$  度, 正北为  $0$  度, 正东为  $90$  度, 正南为  $180$  度, 正西  $270$  度; `orientation1`:  $y$  轴和  $y$  轴在水平面上投影之间的角度, 即  $y$  轴与水平面的夹角, 范围  $-180 \sim 180$  度; `orientation2`:  $x$  轴和  $x$  轴在水平面上投影之间的角度, 即  $x$  轴与水平面的夹角, 范围  $-180 \sim 180$  度。

有了方向传感器的数据, 我们就可以利用方向传感器数据和相对于手机的加速度数据来计算相对于地面参照系的加速度。将相对于手机坐标系的 3 个加速度相对于地面参照系的 3 个方向分解, 再将分解后的加速度分量根据地面参照系的 3 个方向进行相加合成。得到相对于地面参照系的加速度。以地面参考系为参照, 将有利于应用开发。

### 3.3.2 GoogleMap 及 GPS 应用开发

在位置服务方面上, Android 平台上能很好的利用 `MapActivity` 和 `MapView` 直接调用 `GoogleMap`, 还可以进行地图的缩放、查找、定位和跟踪等各项功能。类 `Location` 包含了所在地的经纬度坐标、高度、方向以及

精确度等信息，可以利用 LocationManager 和 LocationProvider 两个类得到，而 Geocoder 类则提供了正逆向编解码即位置名称和经纬度坐标的相互转换<sup>[11,12]</sup>。

### 3.3.3 二维码 QRCode 应用

在二维码的名片识别应用上，第三方的类库 SwetakeQRCode.jar 包给了我们很好的实现。我们只需要输入字符串信息，利用 com.swetake.util.Qrcode 类就能很方便的转换成产生 QR 图片文件的二维数组，然后再创建 Bitmap 进行灰度化等图像处理。图 3 就是 QR 码版本 7 符号为例的结构图<sup>[4]</sup>。

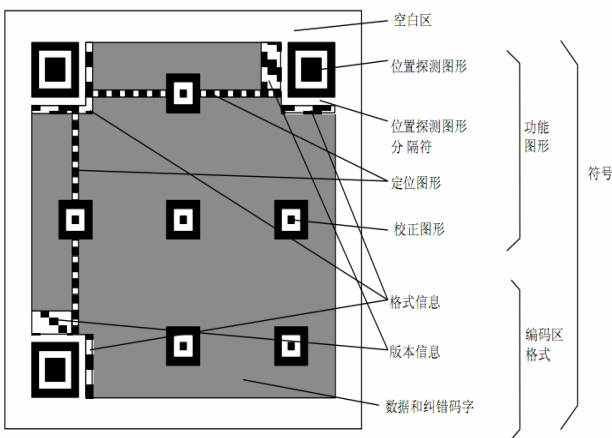


图 3 QR 码结构图

### 3.3 实验过程

#### 3.4.1 传感器解决方案的实现过程

(1) 作为快速匹配设备的加速器应用开发实现过程如下（流程如图 4 所示）：

① 初始化：在这个过程中，完成两个方面的初始化工作，数据管理上获取硬件设备相关参数和以及设定相关碰撞峰值等参数，此外还建立一个状态机，用于监听设备在运行过程中的各种状态；

② 连接阶段：向服务器发送连接请求，若连接成功就向服务器注册服务，发送 GPS 位置以及设备标识符等数据；

③ 监听碰撞：注册成功后，设备便开始一直监听，并采取传感器数据进行逻辑判断是否达到碰撞峰值；

④ 匹配碰撞：此阶段服务器要完成接受设备发送的时间戳等数据进行逻辑判断哪两个设备发生碰撞；

⑤ 交换数据：两个客户端之间的数据通过服务器中转交换数据。

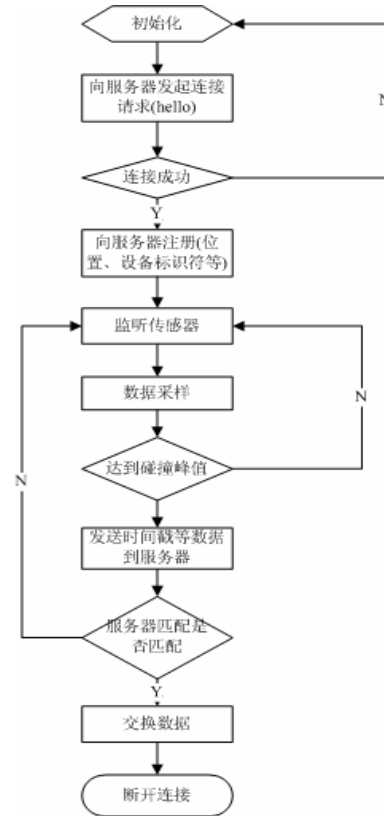


图 4 碰撞流程图

需要注意的是，在整个连接过程中，客户端一直向服务器发送心跳包以保障网络连接的可靠性，将有可能发送的网络异常以友好的形式通知用户。在第④步服务器进行匹配时，首先从 GPS 位置数据进行空间的筛选，再从时间戳进行时间的筛选，从重力加速度 xyz 数组进行更精确的筛选，允许一定的误差存在的。通过实验人为测试每个决策值，确定相关阈值，GPS 的精度 50 米内，时间不超过 5 秒，xyz 数组的角度不超过 0.2 度，匹配的效率  $O(n^2)$ 。

碰撞情况判断算法如下：

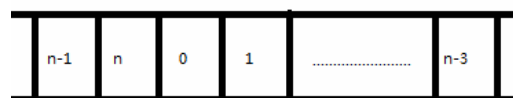


图 5 碰撞算法采用的数据结构

① 每次获取到 x, y, z 数据和前一次的缓存数据进行运算，计算得出一个颠簸情况 bv，即传感器数值变化度 cv 乘上灵敏度决策值 lm。  $Cv = abs(x1-x2)$

$+abs(y1-y2)+abs(z1-z2)$ ,  $Bv = cv * lm$ 。

② 在固定容量为  $n+1$  数组  $cache$  中, 环形地缓存了前  $m$  次的颠簸情况  $bv$ , 和发生的时间  $time$  数据。如上图所示该缓存作用的数组。

③ 判断发生碰撞, 需要判断之前一段时间中手机是否处于平稳状态, 方法是获取环形缓存  $cache$  里面非空且前  $m$  ( $m < n+1$ ) 个  $bv$  和的平均值, 如果小于最大平稳阈值, 则处于平稳状态。如上图, 判断 1 这个点是否发生碰撞, 需要确定  $(n-3,0)$  这个集合是否处于为稳定状态。

④ 如果该  $bv$  大于最小的颠簸值  $minBV$ , 而且手机在前一段时间处于平稳状态, 则可以确定手机发生了碰撞。

(2) 作为接近识别传感器的蓝牙应用开发

针对本应用中建立起机会网络中的每一个存在唯一标识的设备, 将完成以下流程:

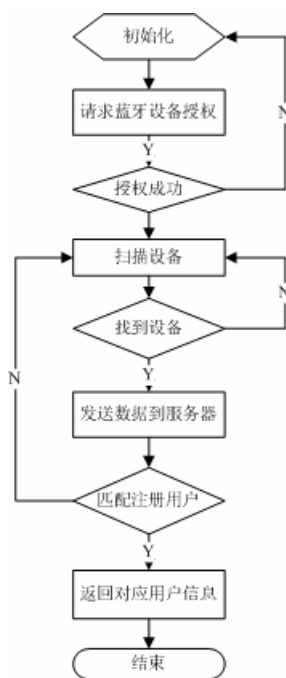


图 6 蓝牙设备连接流程图

当设备连接后就可以进行数据交换操作, 如名片交换、动态更新、聊天功能等。

3.4.2 GoogleMap 及 GPS 实现过程

(1) 显示地图只需下面三个步骤即可:

① 创建  $MapView$ , 并绑定从  $Google$  申请到到  $Android Map API Key$ ;

② 实现  $MapActivity$ , 管理生成的  $MapView$ , 并由  $MapController$  进行控制地图显示的地点以及比例尺;

③ 自定义图层  $Overlay$  在地图上加载代表用户属性的图标文字信息;

(2) 定位用户信息, 借助  $GPS$  定位系统, 使用下面的  $API$  可以轻松完成:

① 通过  $getSystemService()$ 取得  $LocationManager$  实例, 并通过它获得当前可以利用的  $LocationProvider$ ;

② 注册周期性的更新视图的监听器  $location Listener$ ;

③ 实现  $locationListener$  中的回调函数,  $on LocationChanged(Location)$ , 坐标改变时, 系统自动调用此方法, 进而获得记录诸如经纬度高度等位置信息  $Location$ ;

④ 通过  $Geocoder$  正逆向编解码, 得到用户友好性的地理位置名称。

3.4.3 二维码 QRCode 的实现过程

作为身份名片功能的二维条形码识别, 其实现过程如下:

① 用户根据自己的身份  $ID$  通过服务器生成唯一标识的二维条形码, 在使用时显示该条形码图片在自己手机上。

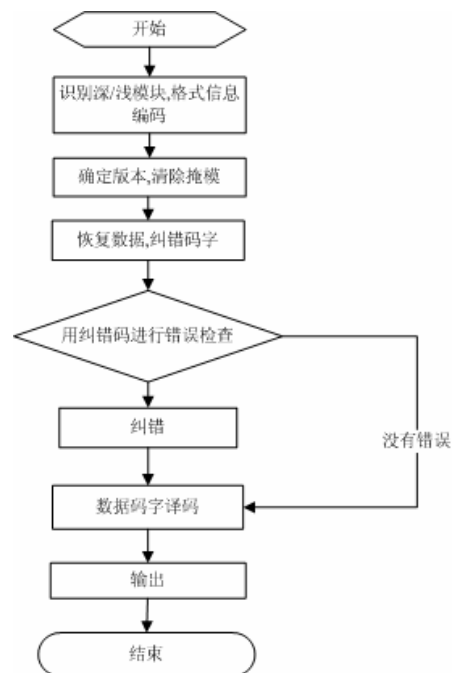


图 7 QR 码解码过程图

② 让其他用户通过摄像头获取该图片，然后手机发送该 bitmap 图片和主机 id 到服务器，在服务器上对二维码进行解码。查询数据库中的对应信息，返回给设备给请求的用户，实现数据的传递。

③ 为了加快 QR Code 译码的速度，采用的分辨率为 160\*120，但可能拍照下来的图片模糊，导致辨识度不高，尤其在条形码包含文字过多时，分辨率更低，解决的方案有两种：方案一是修改相机预览的分辨率为 320\*240 较高的画质精细解码，以牺牲时间换取解码成功率；方案二是将图片以黑白色作为调色板，将图片进行灰度处理、二值化等多种图像处理后再进行解码。解码过程如图 7 所示<sup>[7]</sup>。

### 3.4 实验结果呈现与评价

#### 3.4.1 传感器实验效果

##### (1) 传感器的数据采集

在此方面，我们查阅相关书籍和资料，从理论上进行了推论以及根据现有资料进行真机实验。以下实验数据取自 Moto ME525 android2.1-update1 系统。波形图如图 8 所示：横轴为每个采样点的索引，纵向为数据的变化率。

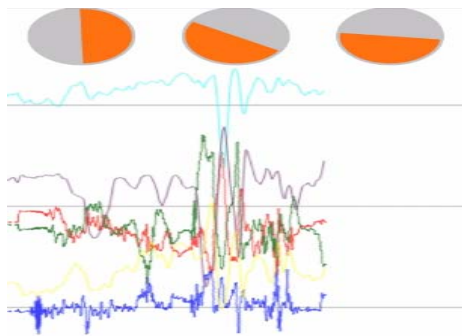


图 8 Android 系统传感器数据采集图

图 8 中，最顶上三个圈是 Sensor.TYPE\_ORIENTATION（方向传感器）获得的数据，分别是方向的坐标值 X, Y, Z 的变化。图中颜色为 RGB(255, 64, 64), RGB(64, 128, 64), RGB(64, 64, 255) 分别是 Sensor.TYPE\_ACCELEROMETER（加速度传感器即重力感应器）产生的数据在 X, Y, Z 轴上产生的变化，颜色为 RGB(64, 255, 255), RGB(128, 64, 128), RGB(255, 255, 64) 分别是 Sensor.TYPE\_MAGNETIC\_FIELD（地磁传感器）产生的数据在 X, Y, Z 轴上产生的变化。

我们将对实验数据进行过滤提取为关键数据，可以从两方面进行提取数据：1、宏观上从手机方向的变化序列提取；2、从具体的数值变化序列提取。目前实现的方案是在手机端按照一定规则采集数据，包括 GPS，时间点，硬件标识符。进而进行分析。

首先同步各个设备与服务器的时间差，计算出最小的网络延时。然后采集加速度的变化数据，进而判断是否发生碰撞，记录发生碰撞的时间，发送到服务器进而匹配出成对的设备。

然后启动两台设备，开始碰撞，寻找最匹配的两台设备，比较了 GPS 位置和时间差。实验结果是最好的时间差为 1.29 秒。

在服务器返回结果时，为这两台设备建立连接，提供数据缓存，这两台设备就可以开始交换名片数据。效果如图 9 所示。

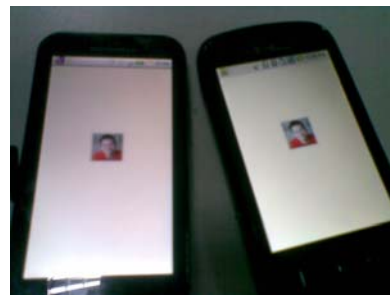


图 9 碰撞初始化

建立连接后，数据可以从一个设备“拖动”到另一个设备。右边设备拖动一个名片，然后根据上文提及的相对地面的加速度方向，判断右设备名片消失的边界，以及左设备出现该名片的边界。从右设备拖动名片或者其他文件图标，当拖动到右设备的边界，左设备开始出现该数据图标，说明数据交换成功。



图 10 准备交换数据



图 11 成功交换数据后

### (2) 蓝牙感应的实验效果

在蓝牙感应上面，通过多台备有蓝牙设备手机的实验，一台手机设备发现另一台设备的平均时间为 2.7s。而有效感应的距离有 7 至 8 米左右，通过发送位置信息到服务器进行运算，进而进行多人位置感应，从周围人到周围人的周围人，这种使用机会网络的机制的方式会使感应到的人数的增多。

如图 12 所示，初始化后扫描效果：出现 3 个蓝牙设备。而图 13，在通过机会网络中“存储-转发”机制，能够发现第 4 个蓝牙设备。



图 12 蓝牙初始化扫描结果



图 13 机会网络“存储-转发”机制扫描结果

### 3.4.2 GoogleMap 及 GPS 实验效果

用户登陆后，从省流量的角度考虑，我们默认将用户好友的地理位置信息以列表的形式展现出来，如图 14；当然用户也可以按照自己的意愿切换至地图模式，用户想知道好友确切的状态等信息时，可以点击标识有好友姓名的气泡，展现出确切的状态，如图 15 所示。



图 14 好友列表展示

### 3.4.3 二维码的实验效果

在二维条形码的实际操作中，发现在手机照相获取另一个手机或者纸上的二维条形码时，图片分辨率较低时，解码转化效率和成功率降低，尤其在条形码包含文字过多时，分辨率更低，对上述所讲的两方案进行了实验，从实验效果上看，成功率有所提高。进行自动对焦的时候，先寻找选取了 3 个位置探测图形，在这个过程中，能够进行多次的采样，如果能得到标点则为要采集的图形，确定好标点的时候进行图形的扫描。进行完二值化的过程后，再依次横向扫描，得到数值。



图 15 好友地图展示

如图 16 所示,图中为根据文字名片生成二维码名片,图 17 为解码效果。实际测试过程中,手机离图片距离要适中才能尽快得到解码结果,再进行相应的数据处理。



图 16 生成二维码名片



图 17 解析二维码名片

#### 4 结语

移动 SNS 是当前研究的一大热点,在现实生活中有很大的应用价值,本文通过对智能手机终端带有的重力加速度传感器和蓝牙、WiFi 等技术手段建立起面向现实的社交应用系统,通过交互实时数据,拉近人与人之间的距离,有很强的实用价值。为了更好的提

高用户的友好性,以及通过 WiFi,蓝牙建立机会网络,下一步的研究方向提高机会网络的数据传输效率,以及机会网络节点的数据路由上做进一步研究,以提高整体性能。

#### 参考文献

- 1 陈飞翔,李华,等.面向 LBS 的移动空间信息服务研究.计算机工程与应用,2008,44(13):217-219,224.
- 2 王玉祥,乔秀全,等.上下文感知的移动社交网络服务选择机制研究.计算机学报,2010,33(11):2126-2135.
- 3 耿东久,索岳,等.基于 Android 手机的远程访问和控制系统.计算机应用,2011,31(2):559-561,571.
- 4 熊永平,孙利民,等.机会网络.软件学报,2009,20(1):124-137.
- 5 牛建伟,周兴,等.一种基于社区机会网络的消息传输算法.计算机研究与发展,2009,46(12):2068-2075.
- 6 GB/T 18284-2000 neq ISO/IEC 18004:2000.国际快速响应矩阵码.
- 7 Y.Swetake.QRcode <http://www.swetake.com/qr/index-e.html>
- 8 Google.zxing.<http://code.google.com/p/zxing/>.
- 9 杨丰盛.Android 应用开发揭秘.北京:机械工业出版社,2010.277-280.
- 10 Google.android API<http://developer.android.com/reference/android/hardware/SensorEvent.html>
- 11 郭宏志.Android 应用开发详解.北京:电子工业出版社,2010.287-304.
- 12 杨丰盛.Android 应用开发揭秘.北京:机械工业出版社,2010.283-297.