

基于 Android 平台的通讯帮手的设计与开发

唐敏

(安徽邮电职业技术学院 合肥 230031)

摘要 Android是由 Google 推出的源码公开的智能手机操作系统,从推出至今得到越来越多的人的研究和使用。通过搭建 Android 应用开发平台,完成了 Android 应用程序的开发以及从总体设计到详细设计的工作。设计了一款基于 Android 的通讯帮手软件,它主要实现了手机联系人的分组存储、联系人资料的详细存储以及快捷的电话呼叫、短信发送、邮件发送功能。应用程序界面都通过 Android UI 设计完成,通过虚拟机调试分析直至最后编码通过。

关键词 智能手机操作系统,移植,Android UI

中图分类号 TP311 **文献标识码** A

Designing and Developing the Communication Helper Based on Android

TANG Min

(Anhui Vocational College of Posts and Telecom, Hefei 230031, China)

Abstract Android is an open source smart phone operating system, which is launched by Google. More and more people spend time in researching and using it. The paper built Android application development platform, and realized Android application program. The work is from overall design to detailed design. The design of the communication helper software is based on Android. It has the function of grouping the contacts stored in the phone, storing contact detail information, and quick phone calls, SMS, e-mail sending. The interface design is completed through the Android UI, and the program is debugged by virtual machine until the code is passed.

Keywords Smart phone operating systems, Transplantation, Android user interface

Android 是 2007 年底由 Google 推出的源码公开的智能手机操作系统,从推出至今已经得到越来越多的人的研究和使用。相关调查数据显示,Android 在 2011 年第一季度占领智能手机系统老大的位置。2011 年 5 月 11 日,第三届 Google I/O(开发者大会)在旧金山举行。此次开发者大会上,Google 在 Android 上的创新给大家带来了惊喜:Android 3.1 的新功能可以支持外接游戏手柄和有趣的面部识别,Android@Home 可让家里所有可连接的物体与 Android 相连^[1]。由此可见,Android 已经开始向物联网技术方面发展。如果继续朝物联网方向研究,Android 的市场前景不可估量。本文提出了一种智能通讯帮手系统的设计实现方法,以供开发者参考。

1 Android 操作系统体系结构介绍

Android 采用了分层架构的思想。表 1 是 Android 的体系结构图,它分为 4 层,从顶层到低层分别是应用程序层、应用程序接口层、系统运行库和 Linux 核心层。Android 是和一系列核心应用程序包一起发布的,这些核心应用程序包括 Email 客户端、信息程序、日历、地图、浏览器功能、联系人管理等,所有的核心应用程序均使用 Java 高级语言编写。开发人员在遵循 Android 应用程序框架的安全性限制条件下,可以访问核心应用程序所有的 API 函数。系统运行库层可以

分为单独的两个部分,一部分是 Android 函数库,这些函数库主要是一些 C/C++ 代码库,函数库与应用接口层配合;另一部分主要是 Dalvik 虚拟机,虚拟机是为调试方便而设计实现的。Linux 内核层相当于系统内核层,是 Android 平台中硬件与软件之间的纽带^[2,3]。

Android 体系结构中 4 层分布很合理。层与层之间的功能相互独立,但是上一层的实现都依赖与下一层的支持,下一层也为上一层的实现奠定了基础,各层提供固定的 SAP (Service Access Point),符合高内聚、低耦合的特点。

表 1 Android 操作系统体系结构

应用程序和小部件
应用程序接口
类库 Android 运行环境
Linux 内核和驱动
硬件

2 Android 应用程序开发组件

一个 Android 应用程序,通常由 Activity、Service、ContentProvider 和 BroadcastReceiver 4 组件构成^[4,5]。

2.1 Activity(活动)

Activity 是最基本的 Android 应用程序组件。一个 Android 应用程序可以被看作一组任务,每一个任务称作一个 Activity。在应用程序中,一个 Activity 通常就是一个独立的

本文受 2011 年度高等学校省级优秀青年人才基金项目(2011SQRL189)资助。

唐敏(1976-),女,硕士生,讲师,主要研究方向为嵌入式技术。

屏幕,用于表现数据和与用户交互,每个屏幕可以有一个或多个组件布局。每一个 Activity 又是一个单独的类,每一个这样的类都从 Activity 基类继承而来。一个 Activity 可以切换到新的 Activity,也可以返回前一个 Activity。Android 通过使用 Intent 这个特殊的类来实现 Activity 的切换。

2.2 Receiver(服务)

服务是具有一段较长生命周期且没有用户界面的程序。它可以运行在它自己的进程,也可以运行在其他应用程序进程的上下文(Context)里面,这取决于其自身的需要。其它的组件可以绑定到一个服务(Service)上面,通过远程过程调用(RPC)来调用这个方法。典型的例子就是音乐播放器。在一个音乐播放器程序中有多个活动,其分别实现了歌曲的选择和播放等功能。然而,音乐重放这个功能并没有对应的活动,它是通过服务来实现的。媒体播放器这个活动会使用 Context.startService()来启动一个服务,从而可以在后台保持音乐的播放。同时,系统也将保持这个服务一直执行,直到这个 Service 运行结束。

另外,还可以通过使用 Context.bindService()方法,将其连接到一个服务上(如果这个服务还没有运行将启动它)。当连接到一个服务后,可以通过服务提供的接口与它进行通讯。就媒体播放器这个例子而言,我们还可以进行暂停、重播等操作。

2.3 Content Provider(内容提供商)

Content Provider 可以实现不同应用程序之间的数据共享,它提供了标准的访问接口以便其它应用程序对数据进行访问和操作。典型的例子就是使用内容管理器来访问联系人列表。你的应用程序也可以使用其它程序通过内容管理器提供的数据,同时你也可以定义你自己的内容管理器来向其它应用提供数据访问服务。

2.4 Broadcast Receiver(广播接收器)

在 Android 中提供了 BroadcastReceiver 组件来响应一个外部的的事件,当一个 BroadcastReceiver 被触发后,系统就会通知相应的程序。例如:当电话呼入、数据网络可用、或者到了晚上时,BroadcastReceivers 不能显示 UI,它只能通过 NotificationManager 来通知用户。BroadcastReceivers 既可以在 AndroidManifest.xml 中注册,也可以在代码中使用 Context.registerReceiver()进行注册。当这些事情发生时,系统会在需要的时候启动你的应用,并在必要情况下触发 BroadcastReceivers。还可以通过使 Context.sendBroadcast()将它们自己的 Intent broadcasts 广播给其它应用程序。

3 应用程序开发环境构建

硬件平台 Mini2440 是广州友善之臂有限公司开发的一款 ARM9 学习板,微处理器采用的是 Samsung S3C2440,是业界领先的 32 位 RISC 微处理器。

对于应用程序开发,笔者选择在 Windows XP 下面完成。Android 应用层主要代码是由 Java 语言编写的,对 Java 开发比较熟悉的都知道 Eclipse 集成开发环境,提供了强大的插件功能,可以方便地支持第三方软件的开发,这里选择 Eclipse(版本 3.5)+ADT 插件。软件开发包选择最新的 Android SDK 版本。当然为了支持集成开发环境,还需要安装 jdk,因

为是 Java 开发,Eclipse 开发环境需要 JRE 的支持,如果没有 Java 运行时环境的支持,Eclipse 就不能正常工作,会显示出错。

开发的应用程序要在开发板上测试。我们要进行 Android 操作系统在开发板上的移植工作,具体移植过程请参考参考文献[7];同时还要建立交叉编译环境为应用程序下载到开发板测试做准备。本文采用 Google 自己提供的工具链 Bionic Lib,可以从 <http://android.kernel.org/pub/android-toolchain-20081019.tar.bz2> 下载到,这里就不再详述。

4 通讯助手的设计

4.1 通讯助手的总体设计

手机作为通信设备,方便大家沟通交流。大家在使用手机的时候,通讯簿是必不可少的工具。本文设计的这款通讯助手,在让用户不必去记手机号码等联系人相关信息的基础上,进行个性化的设计,可以提供方便快捷的使用通道,让用户充分享受智能手机带来的舒适与便利。这为同类软件的设计提供了一个很好的参考。

Android 通讯助手主要是借助于 Android 提供的应用程序开发接口设计完成的,Android 通讯助手里面的数据采用的是系统自带的 SQLite 数据库,所有信息都保存在手机的 data 目录的相应软件包中。通讯助手的主要功能是保存所有用户的信息和资料,包括联系人姓名、电话号码、头像、邮箱地址、生日、联系人描述等信息。

其次,Android 通讯助手提供了人性化的界面,可以通过单击界面图标,实现添加联系人、创建用户组等操作;同时 Android 通讯助手还提供了快捷的通信通道,通过单击形象图标,实现发送短信、发送邮件、呼叫号码等功能。图 1 为通讯助手功能结构图。

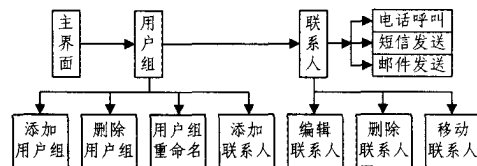


图 1 通讯助手功能结构图

4.2 通讯助手的详细设计

在进行通讯助手的详细设计分析时,主要以通讯助手编辑联系人模块进行分析说明。编辑联系人模块主要用到了 Android UI 设计,下面的部分代码是布局 XML。

```
<? xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <ScrollView android:layout_width="fill_parent"
        android:layout_height="fill_parent" android:layout_weight="1">
        <LinearLayout android:layout_width="fill_parent"
            android:layout_height="wrap_content">
```


叫、短息发送、邮件发送的快捷方式,如图4,即通讯帮手用户组展开图所示。



图3 通讯帮手主界面图

图4 通讯帮手用户组展开图

通过单击图4中所示的 sms 图标,可以直接进行短息发送,系统会自动载入当前联系人的电话号码,在界面空白处,填写短息内容。

关于 Android 通讯帮手用户分组和用户联系人的相关设计,首先利用主界面的类包含的 onCreateOptionsMenu()方法创建 menu 菜单的内容,提供添加新用户组、帮助、退出等功能。然后通过 onCreateContextMenu()添加长按用户组和联系人,弹出下拉菜单项,通过 onContextItemSelected()实现相应功能,如编辑联系人、删除联系人等,在此就不再一一赘述。

结束语 自从 Google 推出 Android 智能手机操作系统以来,Android 的用户数目成倍增加,Android 市场占有率翻了几番,这些都显示了 Android 无穷的魅力。在这种趋势下,智能手机制造商纷纷把目光投向 Android,研究和开发它的应用程序,开发人员也成倍增长,可以说 Android 掀起了智能

手机操作系统一个新的浪潮。在搭建 Android 应用开发平台并测试成功后,所有的应用程序界面都通过 Android UI 设计完成,并在虚拟机里进行了测试。应不断增加和完善通讯帮手的功能和界面,使软件使用起来方便、快捷、美观、安全,让用户体会到智能手机的魅力和软件的智能化、人性化的特点。

参考文献

- [1] Google I/O: 再造创新和开放的基因[EB/OL]. <http://www.cww.net.cn>,2011-06-10
- [2] Butler M. Android: Changing the Mobile Landscape[J]. IEEE Pervasive Computing,2011,10(1):4-7
- [3] 丁永明,纪方明. 基于 Android 平台移动学习软件的研究与实现[J]. 数字通信世界,2011,1:66-68
- [4] 李杨,冯刚,李亮,等. 基于 Android 的多媒体应用开发与研究[J]. 计算机与现代化,2011(4):149-152
- [5] 刘枫. Android 智能手机天气预报系统设计及实现[J]. 计算机与时代,2011(4):61-63
- [6] Micro2440 用户手册[M]. 广州:广州友善之臂科技有限公司,2010
- [7] 宋杰,王书菊,曹竹冬,等. 基于 ARM2440 平台的 Android 操作系统的移植[J]. 计算机技术与发展,2011,21(1):66-69
- [8] 勒岩,姚尚郎. Google android 开发入门与实战[M]. 北京:人民邮电出版社,2009
- [9] 余志龙,陈小风,等. Google android SDK 开发范例大全[M]. 北京:人民邮电出版社,2009
- [10] Gramlich N. Android Programming[EB/OL]. <http://andbook.anddev.org>,2008-02-21
- [11] Maruna T, Savic M, Miljkovic G, et al. Google's Android as an application environment for DTV decoder system[C]// 2010 IEEE 14th International Symposium on Consumer Electronics, 2010:35-44

(上接第 572 页)

由图2不难发现,这种算法对单一背景下的目标跟踪也是非常准确的(如第2~100帧),但在场景发生复杂变化时(如第110~129帧)跟踪会出现局部偏差;借助于目标模式自适应更新算法,目标短时间被遮挡的现象对跟踪结果影响不大(如第130~135帧),但遮挡时间过长或目标与背景相似度过高时,算法会出现明显的跟踪偏差(如第240~300帧),随机关子分布在更大的分散区域,目标估计准确性明显降低。

另一方面,由于增加一种纹理特征导致算法计算量大增,算法的耗时也明显增加了。

结束语 为克服只有单一颜色特征进行目标跟踪鲁棒性不高的问题,提出了融合颜色和梯度方向特征的粒子滤波跟踪算法。该算法利用特征融合的粒子重要性评判模型和目标模式自适应更新模型,提高了算法对复杂变化的自适应能力。实验与分析表明,所提算法能比较可靠地实现复杂场景下的目标跟踪。但是,由于算法复杂性的增加,导致跟踪的实时性降低,因此,如何设计融合多特征的粒子滤波跟踪并行快速算法,增强算法的自适应性仍是值得深入研究的课题。

参考文献

- [1] Comaniciu D, Ramesh V, Meer P. Real-time tracking of non-rigid

objects using mean shift[C]// Computer Vision and Pattern Recognition. Hilton Head, SC, USA: IEEE Computer Society, June 2000:142-149

- [2] 李远征,卢朝阳,高全学,等. 基于多特征融合的均值迁移粒子滤波跟踪算法[J]. 电子与信息学报,2010,32(2):411-415
- [3] 尹宏鹏. 基于计算机视觉的运动目标跟踪算法研究[D]. 重庆:重庆大学,2009
- [4] 张树春,胡广大. 跟踪机动再入飞行器的交互多模型 Unscented 卡尔曼滤波方法[J]. 自动化学报,2007,33(11):1220-1225
- [5] 段其昌,周奇,段盼. 结合水平集和粒子滤波的人脸轮廓跟踪[J]. 光电工程,2010,37(3):117-121
- [6] 曾伟,朱桂斌,陈杰,等. 多特征融合的鲁棒粒子滤波跟踪算法[J]. 计算机应用,2010,30(3):643-645,656
- [7] 常发亮,马丽,刘增晓,等. 复杂环境下基于自适应粒子滤波器的目标跟踪[J]. 电子学报,2006,34(12):2150-2153
- [8] 王书朋,姬红兵. 用于目标跟踪的自适应粒子滤波算法[J]. 系统仿真学报,2010,22(3):630-633
- [9] 刘先省. 基于粒子优化的多模型粒子滤波算法[J]. 电子学报,2010,38(2):302-307
- [10] 牛长征. 复杂背景下视频运动目标跟踪的研究[D]. 北京:北京理工大学,2010