

基于 Android 系统的多传感器移动体感应用^①

庄翠翠, 李成荣, 韦 玮, 罗杨宇

(中国科学院 自动化研究所, 北京)

摘 要: 研究了多传感器技术在 Android 移动设备上的应用, 提出了基于 Android 传感器应用的实现方法, 使用户与设备发生更为直观的操作和交互. 本文首先介绍了开发环境搭建, 利用 Android 程序框架实现此应用程序, JNI(Java Native Interface)调用数据融合相关函数的步骤, 以及采用 Android NDK(Native Development Kit)生成共享库的过程. 然后阐述了利用 Android 传感器和动态调用数据融合算法获得数据开发指南针, 水平仪, 动态立方体的方法; 最后描述了开发的 MFC 服务器端软件通过 Socket 进行实时位置坐标数据的传送和绘制轨迹信息的功能. 实验结果表明, 该系统实现了所需展示的性能和应用功能, 并可扩展到移动设备的其他应用程序中.

关键词: Android; JNI; Socket; 多传感器; OpenGL

Mobile Somatic Application of Multi-Sensors Based on Android System

ZHUANG Cui-Cui, LI Cheng-Rong, WEI Wei, LUO Yang-Yu

(Institute of Automation, Chinese Academy of Sciences, Beijing)

Abstract: This paper studies the application of multi-sensor technology on Android mobile devices, and raises the method of implementations of Android sensor applications, allowing users and equipment more intuitive operation and interaction. First introduced the development environment to build Android application framework to achieve this application, the steps of calling correlation function of data fusion with JNI (Java Native Interface), as well as the process of generating shared libraries with Android NDK (Native Development Kit). Then elaborated the use of the Android sensor and the methods of development of compass, spirit level, dynamic cube; at the same time develop of the MFC server-side software, implementing the transfer of location coordinate data through Socket and draw the trajectory. Experimental results, which show that the system achieve the required function to demonstrate the application, and can be extended to other applications of mobile devices.

Key words: Android; JNI; Socket; Multi-sensors; OpenGL

随着人机交互的蓬勃发展, 人们试图让交互的应用伴随移动用户在对应的空间中迁移, 该模式脱离了传统的键盘、鼠标, 人们可以随时随地获得服务, 充分利用其周围基础设施中资源进行信息处理. 同时其运行的模块也可以与空间中的其它模块进行交互和协作, 以获得更多的服务. Android 设备的日益普及和移动体验的广泛应用使得利用 Android 平台开发移动体感应用程序显得十分重要, Android 移动体感的功能可以应用到很多的程序中, 如控制个人计算机上的演讲稿翻页、音乐切换等^[1].

Android 移动设备中内置了很多传感器, 包括加速度传感器, 陀螺仪传感器, 方向传感器, 磁传感器等, 但单一的传感器获得数据不能满足所有移动体感应用中所需的姿态航向和位置数据, 特别是对于静态和动态精度要求高的应用, 本文把多个传感器的数据通过 NDK 封装的 C 函数进行融合处理, 并把结果用于开发指南针, 水平仪, 3D 立方体和绘制运动轨迹等应用, 以此来直观体验经算法处理后数据对于个性参数指标的满足程度. 同时将 PC 作为服务器端, 手机端通过服务器端提示的信息设置 IP 地址和端口进行 socket

^① 收稿时间:2013-02-25;收到修改稿时间:2013-04-01

连接,发送位置坐标数据,服务器接受数据并显示,两端同步实现绘制运动轨迹的效果。

1 开发环境搭建

Android 开发应用程序采用 Java 编程语言,在 Windows 系统下,需要安装和配置相应的 Eclipse IDE、JDK、Android SDK 以及 ADT。要解决 Java 程序与 C/C++ 程序互相调用的问题,需要利用 JNI 编写本地代码,并采用 Android NDK 编译本地代码。主要安装的内容包括:

- (1) 下载和安装 JDK,配置 Java 开发环境,测试安装是否成功。
- (2) 下载 Eclipse,然后直接解压缩到本地路径。
- (3) 下载 Android SDK 和 Android NDK,并分别配置解压缩的路径。

2 开发技术与系统实现

2.1 系统总体架构

整个传感器应用开发的框架采用客户端/服务器端(C/S)架构,Android 手机作为客户端,PC 运行的 MFC 作为服务器端,两端通过 Socket 进行通讯。服务器端通过 MFC 提供发送数据,显示数据,轨迹绘制等功能;Android 客户端上通过 NDK 把数据融合算法封装成 .so 库文件,然后通过 SDK 和算法获得的数据信息进行客户端 UI 和应用的开发,如图 1 所示。

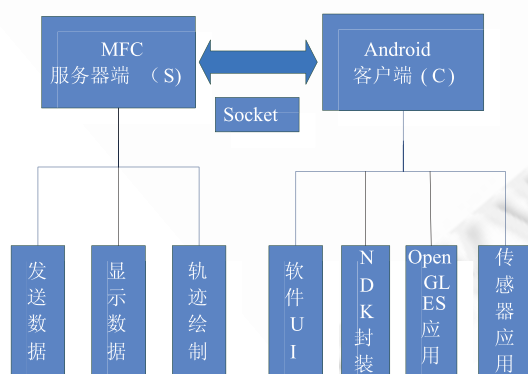


图 1 系统总体框图

2.2 Android 客户端实现

2.2.1 基于 MVC 的 UI 布局设计

MVC (Model-View-Controller), M 是指逻辑模型, V 是指视图模型, C 是控制器。Android 界面部分也采用

了 MVC 框架,在 Android 中,视图层(View)一般采用 XML 文件进行界面的描述,控制层是通过 Activity 控制业务逻辑,模型层(Model)是对数据库和网络的操作处理。

基于本系统的功能需求,在 res/layout/文件夹中新建两个 xml 布局文件 main.xml, track.xml。两个文件都是通过选项卡(TabWidget)来进行界面布局的,通过多个标签切换显示不同的内容,有利于减少显示空间。

同时在界面中通过 onCreateOptionsMenu 和 onOptionsItemSelected 来实现菜单选项功能, onCreateOptionsMenu 中的菜单内容可以通过在 xml 添加或者在函数内部直接动态添加。

2.2.2 多线程实现界面更新和后台运行

为了使得多个工作同时完成任务,提高系统运行效率,采用多线程编程开发技术。由于处理数据融合算法的耗时较长,为了防止 UI 线程阻塞,采用 Thread 线程开启后台运行方式,用 synchronized 处理线程同步问题,即不管任一线程运行到这个方法时,都要检查有没有其它线程正在使用这个方法,发现有使用 synchronized 方法的线程时要等此线程运行结束再运行,否则直接运行。

Runnable 运行时通过 sendMessage 来发送数据更新 Handler,具体方法是首先开辟一个 Message 的对象,然后通过 setData 方法来获得 bundle 对象数据并发送到 UI 线程。此外更新 UI 必须在主线程中完成,Handler 是通过 getData 来获得发送的数据来实时更新动态改变效果^[2]。

2.2.3 NDK 封装 C/C++ 函数

在本系统中,Android 传感器通过 sensorManager 获得加速度计,陀螺仪和磁阻计的数据,经数据融合算法处理后得到姿态和航向数据,即俯仰角(pitch),横滚角(roll),偏航角(yaw),并同时通过算法的中间变量四元数计算获得 x,y,z 轴的位置数据。

为了提高数据的收敛性和运行性能,通过 C 语言编写数据融合算法函数来处理运行结果,然后通过调用动态链接库来获得相应的数据。具体数据融合处理函数为 void ahrsAlgorithm(double*data, double interval, double* result),data 代表加速度计陀螺仪和磁阻计获得的九个数据, interval 代表采集数据的时间间隔, result 代表经算法处理获得的姿态航向和位置的六个数据。

对 C/C++ 函数的调用,Android 是通过 JNI 调用本

地方法,本地方法是以.so 库文件存放的,JNI 是连接 C++和 JAVA 的接口,如图 2 所示。

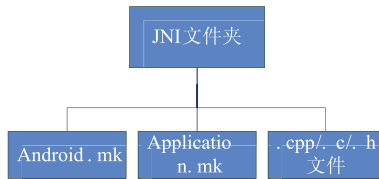


图 2 接口层次框图

(1) 在 JAVA 程序中,首先需要在类中声明所调用的库名称,通过 static 块来加载模块名。此外还需要对将要调用的方法做本地声明,关键字为 native。

(2) C/C++中 JNI 使用方法

JNI 接口函数的命名有一套规则,C 和 C++的命名不同就是在 C++中加上 extern "C",然后加上返回类型,包名,类名和参数。在 C 中,JNI 函数调用由“(*env)->”作前缀,目的是为了取出函数指针所引用的值^[3]。在 C++中,JNIEnv 类拥有处理函数指针查找的内联成员函数。在对返回类型是二维数组的函数里,需要通过返回 object 的方法来实现,相对于基本数据类型和一维数组要复杂,首先要定义一个 jclass,然后通过 FindClass 方法定义数据类型,后再通过定义 NewObjectArray 方式新建 object 数组,通过两次 for 循环来传递数据。

需要注意的是,在接口中获得和定义的参数需要及时释放内存空间,否则当接口被 java 函数多次调用时,就会有异常,导致程序出错。

(3) 配置脚本文件和编译动态库

Android NDK 编译前需要编写两个脚本文件,分别是 Android.mk 和 Application.mk 文件。Android.mk 文件描述了需要 NDK 进行编译的源文件,以及所要形成的组件,Application.mk 文件描述一些关于本应用的辅助编译信息^[3]。

编译 C/C++代码生成动态链接库,运行-> cmd-> 进入项目工程文件夹->ndk-build>项目文件夹生成 libs 文件夹和 obj 文件夹->编译成功,编译结果如图 3 所示。

```
F:\china.mobile.com.SensorAPPActivity>ndk-build
"Compile thumb : AHRS2 <= ahrs_main.c
"Compile thumb : AHRS2 <= matrix.c
SharedLibrary : libAHRS2.so
Install       : libAHRS2.so => libs/armeabi-v7a/libAHRS2.so
```

图 3 编译动态链接库

2.2.4 传感器数据应用开发

为了直观的显示通过数据算法处理获得的方向角,倾斜角,旋转角数据,利用指南针来表现方向角的数据,水平仪表现倾斜角和旋转角的数据。

指南针和水平仪的布局是通过定义 mainview 类来实现的,通过复写 onDraw 函数通过 canvas 绘制图片组件的方式加载显示图片,然后设置 rotate 角度来动态显示指针位置,水平仪则是通过角度转化为位置信息的逻辑算法来动态确定位置,开发效果如图 4 所示。



图 4 指南针和水平仪效果展示

为了把方向角,倾斜角,旋转角的数据集中表现,观察数据之间的影响和同时变化的效果,通过 OpenGL ES 实现 3D 旋转立方体来直观表现。

在 Android 中通常使用 OpenGL ES 来绘制图形,OpenGL ES 为嵌入式系统的一个二维和三维图形库,为桌面 OpenGL 的一个子集。OpenGL ES 定义了一个在移动平台上能够支持 OpenGL 基本功能的精简标准,以适应移动终端显示系统。

在 Android 中我们使用 GLSurfaceView 来显示 OpenGL 视图,该类位于 android.opengl 包里面。它提供了一个专门用于渲染 3D 的接口 Renderer,要实现此接口,需重写三种函数方法: onSurfaceCreated() 方法用于纹理初始化; onSurfaceChange 方法用于设置 OpenGL 场景大小; onDrawFrame() 方法用于绘制不同的纹理填充的图形^[4]。

在 Android 中使用 GLSurfaceView 来显示 OpenGL 视图,该类位于 android.opengl 包里,然后通过 glRotate 来动态显示各个角的变化,开发效果如图 5 所示。

2.3 MFC 应用开发

MFC 是 Win API 与 C++的结合 API,它使得用户在 Windows 下开发应用程序变得容易,隐藏了很多程序开发人员在 Windows 下用 MFC 编制软件时的大量细节,如应用程序实现消息的处理,设备环境绘图等。

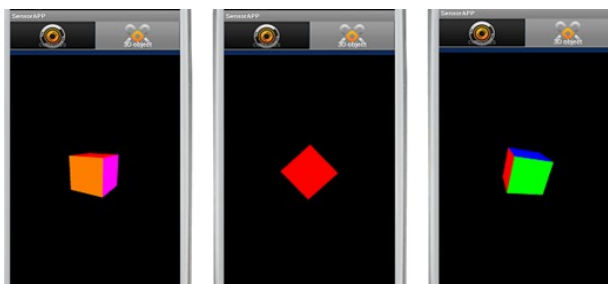


图 5 3D 立方体效果

整个 MFC 应用分为四个部分: 显示信息, 按钮选择功能, 显示轨迹, 发送信息, 通过 MFC 框架来实现按钮控制显示文本和轨迹等功能, 界面如图 6 所示。

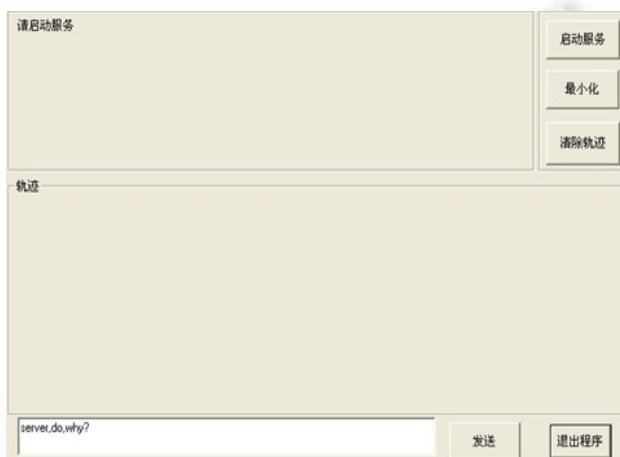


图 6 MFC 应用界面图

2.4 客户端和服务端通信模块设计

Socket(套接字)是一种抽象层, 应用程序通过它来发送和接收数据. 使用 Socket 可以将应用程序添加到网络中, 并与其他应用程序进行通信.

Socket 主要类型为流套接字(stream socket)和数据报套接字(datagram socket). 考虑到数据发送的准确性, 使用基于 TCP 协议的 Socket. 客户端通过 Socket 的连接方法, 服务器端通过 ServerSocket 的连接方法来实现此功能. 点击 PC 端的启动服务按钮, 显示信息区会显示要客户端连接的 IP 地址和端口号, 在手机端输入提示的信息, 点击开始连接, 手机端和 PC 端都会同时显示连接成功的信息, 效果如图 7 所示.

数据融合算法中的四元数在进行坐标系的转换后经过两次积分可获得传感器的坐标位置信息, 为了直观显示这些位置数据, 使用绘制运动轨迹的方法来表

现这些数据的变化. 同时把数据发送到服务器端, 在服务器端同步绘制运动轨迹, 效果如图 8 所示.



图 7 两端通信连接

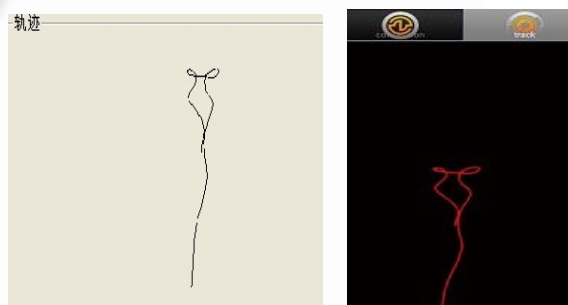


图 8 两端数据通信同步绘制轨迹

3 结语

整个系统是由手机端通过 Android SDK 进行开发, 同时利用 Android NDK 封装 C/C++ 接口函数供 java 调用, PC 端通过 VC++ 的 MFC 来开发应用程序, 两端相互发送信息和接收数据实现的. 系统的实现充分说明了在 Android 系统下进行多个传感器的数据融合, 利用 NDK 调用 C/C++ 函数接口和实时数据通信是可行的, 对增强移动体感的应用有很大帮助, 同时利用传感器进行开发可以更好地扩展到其他人机交互应用中. 随着硬件配置和软件开发技术的提高, Android 系统也将发展更为迅速.

参考文献

- 1 韩迪, 潘志宏. 基于 Android 移动设备传感器的体感应用. 华南理工大学学报(自然科学版), 2012, 40(9): 75-80.
- 2 王宏彬, 李灯熬, 赵菊敏, 熊玉虎. 利用 OpenCV 实现在 Android 系统下的人脸检测. 软件, 2011, 32(12): 4-8.
- 3 金智义, 张戟. 基于 Android 平台的串口通信实现. 电脑知识与技术, 2011, 7(13): 2983-2985.
- 4 杨丰盛. Android 应用开发揭秘. 北京: 机械工业出版社, 2010. 484-501.