手势识别器的设计

Design of Gesture Recognition

■ 哈尔滨工业大学深圳研究生院 刘钊 原志杰 徐维昌 宋翔 张世琦 黄棋波

引言

手机、MP3播放器、硬盘播放器、数码相机、PDA等设备都是通过导航键对其进行控制的。目前比较流行的导航键控制方式有四维键、摇杆，这是最常见的两种导航键，此外还有一些手机上有很新颖的导航键设计，例如：LG-KG70的滚轴键、LG-KE608的转盘设计、索爱W830的触摸式、多普达D802的飞梭滚轮等等。这里，我们用加速度传感器设计一种看不见的导航键来代替四维键的功能，这种方案更能满足消费者的好奇心，满足消费者追逐时尚的需求。

手势识别的控制原理

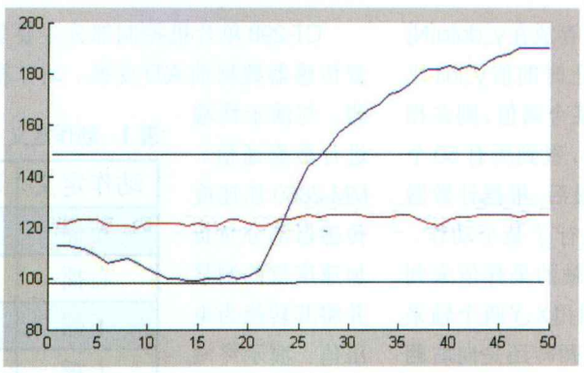
本系统利用三轴加速度的值来判定对物体运动预定义的六种姿势。首先，分别对三个轴采样，每个轴各获得50个数据；然后，分别对每个轴上的数据进行处理来判定是否发生了预定义的动作。动作定义在下面的部分说明，这里仅用Y轴来说明判定的原理。

获得Y轴上的50个数据存放到数组y_data[N]中，将这50个数求和取平均值。若y_data[N]中数据最大值与最小值之差在一个设定的阈值之内，则认为物体在Y轴向上是没有动作的、静止的，此时更新y_init值为y_data[N]求得的平均值；否则，y_data[N]中数据最大值与最小值之差超出一个设定的阈值，则认为物体是运动的，y_init值不变，仍然为上一次静止状态时的值。

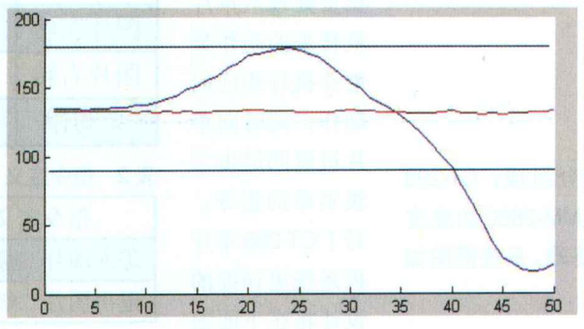
图1和图2是Y轴分别向左、向右摇动时采样得到的加速度抽样值y_data[N]。图中红线代表上一次静止时的采样值，蓝线代表运动时的采样值。图1为向左摇动时的值，可以明显看出加速度的值较静止时有明显的变化，向着增大的方向变。

从图1和图2中可以看出，两条黑线之间的数据很难断定是哪个动作产生的，因为两个动作都可能产生这样的值。所以，利用黑线之外的数据来判定是向右还是向左摇。因为对于这两个动作，黑线之外的数据有明显的差异，数据相差很多。对于图1这些数据大于150，图2中这些数据小于90。因此这些数据至少相差60，可以很容易地将向左、向右的两个动作区分开。

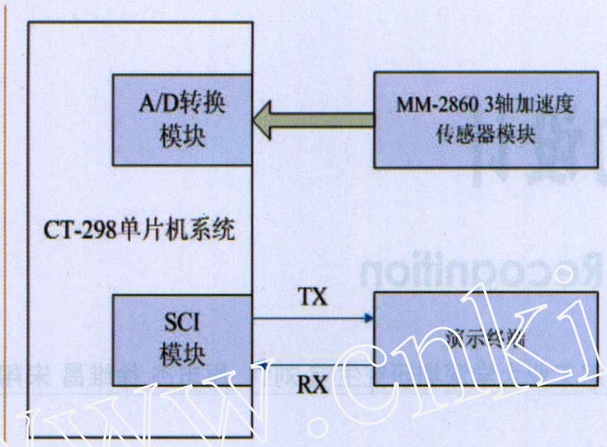
本系统就是根据这样的原理来实现的，首先分别设定向左、向右摇时的阈值和两个计数器；然后，将新



◀◀ 图1 向左摇

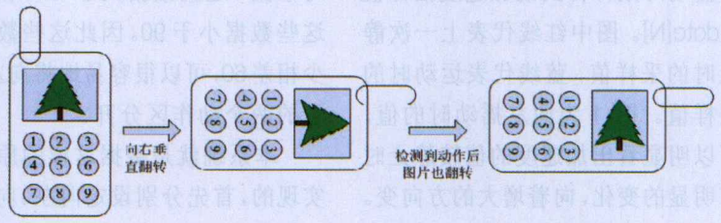


◀◀ 图2 向右摇



▶▶ 图3 系统框图

▶▶ 图4 右摇



▶▶ 图5 图片向左翻转

采样得到的50个值存放在y_data[N]中，将每个值与静止时的值y_init比较，如果数据超出某个阈值，则在相应的计数器上累加，直到所有50个数全部比较完毕；最后，根据计数器值的大小来判定执行了某个动作。

同理，利用Z轴的采样值来判定向上、向下摇，利用X、Y两个轴来判定执行图片向左翻转还是向右翻转，这里略去。

系统描述

系统组成及功能

系统由三个部分组成：CT-298单片机控制部分、MM-2860加速度传感器部分、演示终端，系统框图如图3所示。

CT-298单片机控制部分主要负责传感器数据的A/D变换、动作检测、与演示终端进行数据通信。MM-2860加速度传感器部分负责加速度值的测量并将其转换为电压值。演示终端负责处理从单片机传来的动作类型并执行相应的动作，同时向单片机返回进出二级菜单的指令。对于CT-298单片机系统更详细的设计将在下面的部分介绍，这里首先介绍一下本系统设计的动作姿势及代码，以及与演示系统交互的指令。

预定义的动作

系统设计了六种动作，他们分别是左摇、右摇、下摇、上摇、图片左转、图片右转。图4中的图形象地说明了其中的两个动作，其它的动作与此类似。

系统还为每个动作定义了相应的动作代码，见表1。

当单片机检测到某个动作时会将相应的值赋给变量 type_action，之后通过SCI将type_action的值发送到演示终端。如果用户执行的动作不在这六个动作之中，则将NO_ACTION赋给type_action，表示演示终端不执行任何动作。

系统还定义了单片机与演示终端交互的指令，用于系统在主菜单和二级菜单之间的界面切换，指令从演示系统通过 SCI 传给单片机，单片机接收到之后进入或退出二级菜单，见表2。

这里仅定义了一个二级菜单，即图片菜单，系统还可以定义更多的二级菜单和三级菜单。此外，需

表1 动作定义

动作定义	动作代码	动作宏定义
左摇	0x01	MOVE_LEFT
右摇	0x02	MOVE_RIGHT
下摇	0x04	MOVE_UP
上摇	0x08	MOVE_DOWN
图片左转	0x10	MOVE_TURN_LEFT
图片右转	0x20	MOVE_TURN_RIGHT
无动作	0x00	NO_ACTION

表2 指令定义

指令定义	指令代码	指令宏定义
进入图片二级菜单	0xF1	PICTURE_ENTER
退出图片二级菜单	0xF2	PICTURE_OUT

▶▶ 图6 手机界面

要说明动作 MOVE_TURN_LEFT、MOVE_TURN_RIGHT用于图片翻转，仅在图片菜单中可用，在主菜单不可用。动作下摇执行的是换下一幅图片，动作上摇执行的是退出图片菜单。

演示终端

本系统目标定位在便携式消费类电子产品上，因此在软件应用程序上必须选择具有广泛基础的应用平台。由于近年来J2ME在便携式终端中的应用非常广泛，因此本系统也采用了J2ME平台进行开发。通过比较，选择了J2ME的WTK开发包，它是专门针对移动无线设备而设计的开发包，并提供了一个统一的平台。在WTK的框架下开发出来的java程序可以被众多的移动设备所支持，所以能够有效解决兼容性的问题。

在WTK下我们利用默认的一个手机样机的仿真器DefaultColorPhone进行开发，DefaultColorPhone的样子如图6。

仿真器的外观和操作类似一部移动电话，但是并不代表某个特定的设备，而是提供对其所支持的API的正确实现，每个命令按钮对应着相应的API函数。从图6中可以看到手机的导航键，我们更改了导航键的API函数，使其从鼠标单击触发的方式转换为串口动作代码控制。首先，我们编写了J2ME的串口接收程序，用于接收从单片机传来的单字节的type_action值，然后将type_action以参数的形式传给API，手机根据不同的type_action值执行不同的动作，包括菜单上下翻、进出二级子菜单、图片翻转等。手机动作与type_action的对应关系如表3。

根据type_action的值，在手机界面上产生相应的动作，手机界面发送不同的变化。演示终端的具体实现在下面的内容详细介绍。

硬件描述

系统硬件分为两个部分：CT-298和MM-2860。

CT-298是由MC9S08QG单片机构成、由USB总线电源供电的小型评估板。CT-298上安装有按钮开关、LED灯、蜂鸣器等作为开发的输入输出器件。同时，USB-COM转换电路采用了FTDI公司制造的FT232R，容许单片机与电脑之间通过USB接口进行串行通信。BDM用于代码的烧写及系统的调试。

MM-2860是含有Freescale公司制造的MMA7260Q型三轴小量程加速度传感器的模块，它可以直接安装在CT-298为其设计的插口上。MM-2860的电源是由CT-298上的MC9S08QG8单片机的PTB5端口来控制的，当PTB5端口为L时电源接通。此外，g-SELECT开关是选择传感器灵敏度的开关，使用时将MM-2860插入到CT-298的接口中即可。本系统采用加速度传感器的灵敏度选择为800 mV/g。

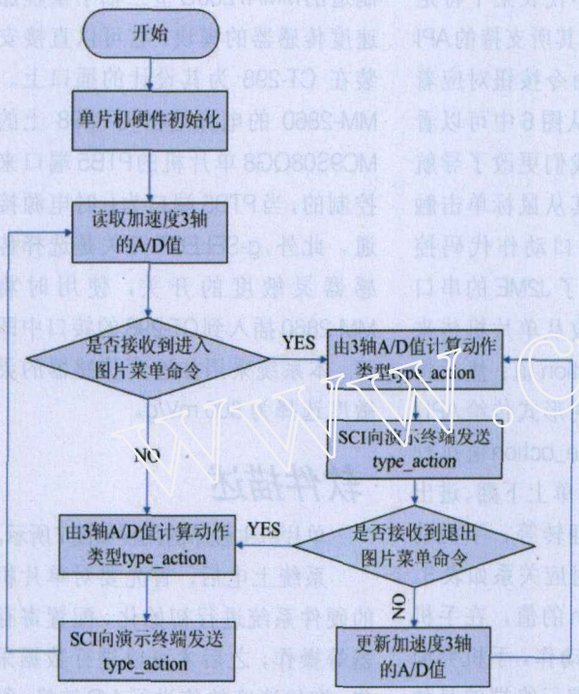
表3 目录与动作代码的对应关系

主菜单目录		图片目录	
手机动作	type_action	手机动作	type_action
上一个菜单	0x04	下一个图片	0x04
下一个菜单	0x08	退出	0x08
进入	0x02	向左翻转	0x10
退出	0x01	向右翻转	0x20

软件描述

单片机主程序的流程如图7所示。

系统上电后，首先要对单片机的硬件系统进行初始化、配置寄存器等操作，之后才可以进行数据采集，将加速度的值进行A/D转换，得到量化的值。图片菜单是二级菜单，如果没有接到演示系统传给单片机进入二级菜单的指令，则单片机一直在主程序运行，不断地采集A/D值、进行动作判定，并向演示系统发送type_action的指令。演示系统可根据接收到的type_action的类型采取相应的动作。如果单片机接收到进入二级菜单的命令，则进入图片菜单，同时也执行类似于主程序的动作判定程序，并不断更新A/D采样值，发送type_action，直到接收到退出二级菜单指令才退出。下面详细介绍一下各个功能模块的具体设置。



▶▶ 图 7 主程序流程图

单片机硬件初始化

单片机系统主要的工作有：将加速度传感器的模拟数据进行 A/D 转换、向演示系统发送 type_action 的动作类型、接收演示系统发来的进出二级菜单的指令、设置采样值，除此之外还需要对系统时钟、外部设备（灯，buzzer）进行配置。根据单片机的主要工作内容选择单片机内部的功能设备，包括A/D转换器、模定时器、串行通信模块（SCI）、内部时钟源模块。

数据采集

系统设置的采样频率为 200Hz，每秒钟分别对X、Y、Z 三轴采样 200 个数据，因此定义了三个大小为N的数组对数据进行缓存，他们是：

```
char x_data[N];
char y_data[N];
char z_data[N];
```

这里N取 50，每 0.25s存取一次，1s钟可以存取 4 次，保证采样率为 200。函数 void acce_meas(void) 负责将采样的数据分别放到这三个数组中，下面是程序的具体实现：

```
for(j=0;j<N;j++)      //采集 N 个数据
{
  for(i=0;i<3;i++)
  {
   if(i==0)
    {
      adc_go(0);          //选择 A/D 信道 0
      x_data[j]= ADC_val_L;  //X 轴
    }
   else if(i==1)
    {
      adc_go(6);          //选择 A/D 信道 6
      y_data[j]= ADC_val_L;  //Y 轴
    }
   else
    {
      adc_go(7);          //选择 A/D 信道 7
      z_data[j]= ADC_val_L;  //Z 轴
    }
  }
  delay();              //延时函数，用来设定采样率
}
```

动作检测

动作检测主要是用获得的三组数据值 x_data [N]、y_data [N]、z_data [N] 来判定当前动作是系统设定动作中的哪一个，如果符合，则发送这个动作对应的预定义值给演示终端，否则发送 NO_ACTION。type_dectction() 用来实现动作检测，其中 type_move 为函数内部变量，用于记录动作代码。type_dectction() 进行动作检测的流程如图 8 所示。

演示终端

J2ME 平台开发环境配置

进行 J2ME 的开发需要从网络上下载免费的开发环境。运行 Eclipse 后进行最后的配置：Window->preferences->J2ME->Platform Component 然后右键选择 Wireless Toolkil，再选择弹出菜单中的 Add Wireless Toolkil，选择刚刚安装的 J2ME Toolkit 的安装路径：C:\WTK22，这样基本配置就完成了。

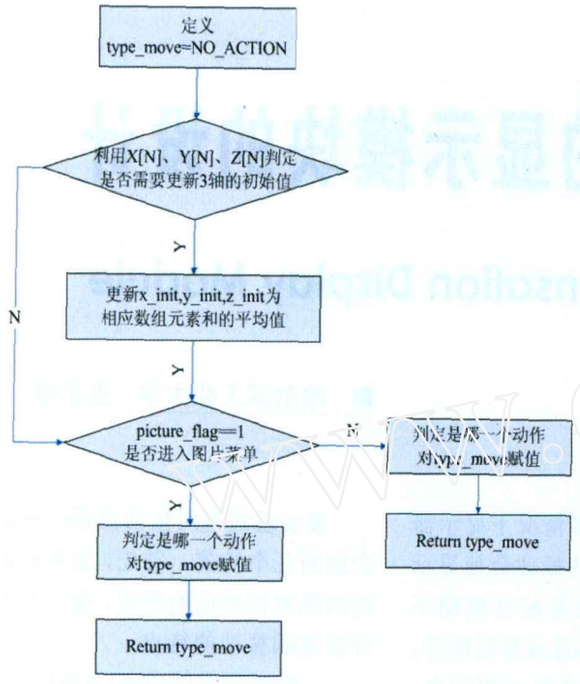
J2ME 的 WTK 开发包是专门针对移动无线设备而设计的开发包，并提供了一个统一的平台。在 WTK 的框架下开发出来的java程序可以被众多的移动设备所支持，能够有效解决兼容性的问题。

软件应用程序

程序分为主进程、串口监视模块和定制化用户界面三大部分，分别基于 J2ME 不同的类库派生而成，三部分之间通过消息机制相互联系，共同构成整个程序的运行周期。具体每个模块所实现的功能参考下节内容。

软件应用程序的组织结构

本软件应用程序在 WTK 的手机仿真器下进行设



▶▶ 图 8 type_dectction()流程图

计，分为 Base 和 display 两个 package。在 Base Package 中包含 MainRoutine.java 和 RS232Port.java 两个 java 文件；在 display Package 中的文件较多，主要实现了 UI 和基本的 key 响应。下面介绍软件部分的实现方法：

· MainRoutine 类

MainRoutine 为程序的入口类，它整合了程序中的所有的对象。MainRoutine 派生于 MIDlet 类，重载实现了 MIDlet 中的 startApp、pauseApp、destroyApp 等方法，并且在 constructor 中加入了 exitCommand 命令，从而实现了手机的关机功能。

可以看出 MainRoutine 实现了程序的入口和退出，并同时标志了程序的基本框架，给具体的功能应用打下了坚实的基础。

· vCanvas 类

vCanvas 继承了 Canvas 类，并增加了 externalMsg 方法。之所以在其中增加 externalMsg 方法，是因为其超类 Canvas 不能动态地响应 key，而我们对 Canvas 的要求是能够通过对串口导入的数据（或按键）得到的 keyCode 做出相应的动作，比如图片的翻转、文本的滚动等等。于是，我们可以通过 vCanvas 派生出一系列的子类，实现我们的具体要求，为将来的开发提供了方便。

· DisplayItem 类

DisplayItem 是基础类，提供了通用的 string 数据组合。该类的构造方法中需要输入 shortText、longText 和 extra 三个 string 参数，这样就保证了每个 DisplayItem 都可以返回三个不同长度的文本信息，在不同的场合使用。其中 shortText 用于 UI 中的标题显示，longText 为 Item 的主体内容，extra 为 Item 的附加信息。DisplayItem 类在本系统中会被例化来描述菜单和子功能的内容，是应用广泛的一个基础类。

· ImageCanvas 类

ImageCanvas 同样继承于基础类 vCanvas，它的主要功能是实现了对图片的浏览及旋转、镜像等基本操作。考虑到目前使用手机浏览图片、拍摄图片的多方向性，如果能够通过加速度传感器自动校正图片的方向，使它自动满足我们想要的方向，那么我们就免除了很多不必要的麻烦。于是我们对图片的浏览功能模块增加了向左或者向右旋转 90° 的功能。这样用户就会惊喜地发现无论他的手机怎样摆放，显示的图片将永远保持与地面垂直。

· ListCanvas 类

ListCanvas 提供了主画面的显示能力，并将用户操作派生到内部聚合成员上面去。具体功能的 Item 将包含在 ListCanvas 之中，每一个 Item 拥有显示在屏幕上的 ShortItem，显示在 detail screen 中的 LongText，还有不被显示的 ExtraText，当然其信息我们也是可以得到的。在 ListCanvas 中，我们可以用“右键”显示 Item 的详细信息，也可以用 SELECT 键来标记每个 Item。

· ItemListCanvas 类：

ItemListCanvas 同样继承了 vCanvas 类，实现了多条目文本浏览的功能，构造方法的输入依次为 Display、Displayable、Font、Title 和 ItemVector。将多条 String 构造为一个 Vector 传递给 ItemListCanvas 后，该类能够提供一个多条目文本实现的用户界面。本系统的“关于本系统”子功能就是通过实例化该类得到的。

结论

本系统采用 Freescale 的单片机 MC9S08QG 和加速度传感器 MMA7260Q 实现了便携式手持设备的手势控制，并采用了开放的 J2ME 平台实现了终端应用程序。系统完整，实用性强，成本低廉，在满足用户基本操作需要的同时，增加了移动设备操作的趣味性和灵活性。同时，加速度传感器 MMA7260Q 较小的体积和独特的节电模式更使其在移动设备的应用中有着极大的优势。可以预见，在未来的手持设备系统中会大量采用类似的手势识别功能，因此具有广阔的市场应用前景。GEC