

研究生课程考试成绩单

(试卷封面)

院 系	仪器科学与工程学院		专业		电子信息	
学生姓名			学号		220213731	
课程名称	嵌入式系统综合设计与实践					
授课时间	2021 年 9 月 — 2022 年 01 月		学时	72	学分	2
简 要 评 语						
考核论题	计步器 APP					
总评成绩 (含平时成绩)						
备注	无锡分校					

任课教师签名：_____

日期：_____

基于安卓手机加速度传感器的计步器 app

摘要：随着社会的发展，人们越来越重视自己的健康，而走路和跑步作为一种方便有效的锻炼方式受到了人们的广泛欢迎，这使得人们对计步器的需求逐渐增加。对于步行的信息可以通过手机上的加速度传感器进行获取，对获得的加速度传感器加以适当的算法就可以计算出步数。本文在该背景下就当下的研究现状进行了介绍，然后据此提出了一种结合动态阈值和时间窗口的峰值检测算法来进行计步，并基于手机自带的加速度传感器，通过软件在 Android 系统上进行了实现。最后本文在不同的条件下对软件进行了测试，发现算法当中存在的一些问题并提出了未来可以采取的解决方案。

关键字：加速度；计步算法；动态阈值；时间窗

1 引言

1.1 研究背景和意义

随着社会的进步和生活水平的提高，人们越来越注重自己的健康。而步行和跑步（以下统称为步行）作为方便又有效的锻炼方式自然受到人们的青睐。医学界也通过多年来对体力活动的形式、时间和强度的大量研究，发现走路是最简单易行且经济有效的运动的方式。它不但不需要特殊技能或仪器，而且容易坚持，不需要花费更多的时间，不需要特殊的场地，受伤的危险很小，在各个年龄段、不同社会背景人群中均可进行。由于步行具有强度低、有节奏、不中断、持续时间较长的特点，所以它能够有效地改善人们的心、肺与心血管的机能，尤其是老年人。因为老年人运动应该要尽量避免因运动过量引发短暂性脑缺血、血压下降、头晕和恶心等疾病，所以中老年人锻炼低强度的步行最适宜。另外，步行还可以有效的预防并治疗肥胖症，体重正常者增加步行量可降低患肥胖症的风险，而对于肥胖症患者可通过步行锻炼来降低体重。因此步行是一种安全的并且老少皆宜的健身方式，未来市场对可精确计算步行的步数、路程和速度等信息的便携式设备的需求量会很大^[1]。

计步器是一种被广泛应用的测量运动参数的设备，其工作原理是感应运动人员的加速度，然后经过一定的算法来计算出运动人员的步数^[2]。步行作为人类活动中最基础、最普及、最重要的运动形式，深入地研究计步器具有极其重要的意义。

1.2 研究现状

早期的计步器是利用加重的机械开关检测步伐，就是一个平衡锤在上下振动时平衡被破坏使一个触点出现通/断动作，再由一个简单的计数器记录并显示步数，这被称作机械式计步器。这种计步器结构简单，价格便宜，但由于它必须以垂直的方式悬挂于腰间，如果位置不对则计步很不准确，目前已趋淘汰。

在 20 世纪 90 年代，微机电系统(Micro-Electro-Mechanical Systems, MEMS)的研究进入了突飞猛进的发展阶段。MEMS 技术带来了两个主要优点，即高灵敏度和低噪声，同时由于 MEMS 技术采用批量生产，有效的降低了传感器的使用成本。随着后期的发展，MEMS 传感器的封装尺寸越来越小，功耗也越来越低。因此 MEMS 传感器正逐渐受到手持设备设计者的青睐。

目前市面上主流都是使用 3D 电子式计步器，即利用 MEMS 惯性传感器测试运动加速度信号，然后根据软件算法来判断步数，计步的精确度较机械式计步器有很大提高。3D 也就意味着设备能够全方位感受人体振动，不受佩戴位置和摆放方位的影响，只要戴在身边就可以计步。

目前市场上的计步器，比较知名的跨国品牌主要有：美国的安康盟(Acumen)、欧西亚(Oregon)，日本的卡西欧(Casio)、欧姆龙(Omron)、西铁城(Citizen)等。国产知名品牌有绿森林、康都等。除了这些成型的计步器外，由于智能机的发展，目前还出现了很多计步软件，下载即可使用^[3]。

从传统计步器到三轴加速度计，人们一直不断追求更为简便和有效的计步工具，智能手机的出现，不单单改变了过去手机只有单一通讯功能的状况，还改变了人们的行为和生活方式。把通过智能手机计步应用采集到的运动步数分享到社交网络，成为了督促人们健康运动的主要动力之一，使用智能手机中的运动应用监测健身状况已经成为了一种新潮流，步行、慢跑更成为了一种时尚。智能手机已经逐步替代了常规的商用计步器装置，并保持了不输于传统计步装置的高精度，通过手机传感器采集到的运动数据实现计步方法成为了一种理想的方式^[4]。

1.3 研究内容

本文对计步检测的相关背景理论知识和现有方法进行了研究，然后借鉴已有的方法设计了一种计步检测方法，并且通过 Android 平台进行了计步系统实现。

2 系统架构

2.1 软件界面介绍

(1)主界面



启动计步器按钮：获取加速度传感器，并注册数值变化监听器，在数值变化时读取数据。
停止计步器按钮：通过注销数值变化监听器，停止计步器计步。

清除计步器按钮：清除当前累计的步数，但是不会停止计步。

倒计时模式按钮：进入子界面，再计步的时候可以倒计时，计时单位是秒。

您当前走的步数：开始计步以后，显示走的步数

初末加速度：所取计步间隔的 xyz 轴初末加速度

触发间隔：所取的计步间隔，单位为毫秒

(2)子界面

设置倒计时(秒)

(请输入大于0的数字)

倒计时(秒): 秒

开始计时

停止计时

清空计时

您当前走的步数为:

0

设置倒计时按钮：在右边输入一个大于 0 的整数值，点击该按钮，设置倒计时的时间，该时间会 显示在第二行的 倒计时(秒): 秒 之中，其单位是秒。

开始计时按钮：点击以后步数清 0，从 0 开始计步且第二行的数字每秒减 1。

停止计时按钮：第二行的数字停止减少。

清空计时按钮：将第二行的数字清 0。

3 软硬件设计

3.1 硬件设计

硬件直接使用安卓系统的手机，本实验使用的是 **Redmi K40**。

3.2 算法和软件设计

3.2.1 计步器的算法原理

(1) 人在行走和跑步的过程中，多处部位都在运动，并产生相应的加速度，加速度与时间大致成为一个正弦曲线，且会在某一点有一个峰值。当出现两次波峰或者波谷的时候，就走了一步。

手机计步器可以分别测得 xyz 三个轴的加速度，其示意图如图 1 所示。除了检测峰值以外，为了抛弃无效的振动，还可以增加时间窗算法，即相邻两步的时间间隔应当在一个范围内，如果在时间窗之外就可以认此步无效。此外还可以设置动态参数（例如动态阈值)来提供自适应的算法^[5]。

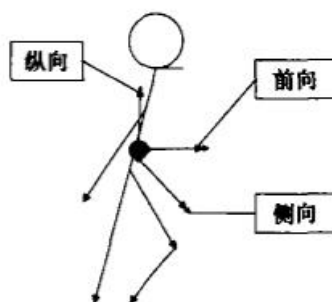


图 1 人体运动时加速度的 3 个方向

(2) 采用的算法介绍如下:

将人步行时候的步行摆动等效为正弦波形, 则加速度波形也可以等效为理想的正弦波形, 然后通过算法来进行计步, 需要满足以下条件以后才将步数+1。

a: 检测波峰和波谷, 检测到两次波峰或者波谷就算一次移动, 但是判断该移动是否是一次步行则还需要满足条件 b 和 c。

b: 检测出现两次波峰或者波谷所间隔的时间, 如果间隔时间 $\geq 300\text{ms}$, 则算作计步间隔。

c: 在满足条件 a 和条件 b 以后, 取波峰绝对值和波谷绝对值的均值作为阈值 (这样可以随随时更新阈值, 以便应对不同的步行状态), 每次得到波谷值以后, 如果波谷的绝对值大于该阈值则步数+1。

将三轴加速度信号进行取模后得到整体加速度 a 计步计算。其计算公式如下:

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

3.2.2 软件设计

(1) 主界面布局代码(activity_main.xml)

对主界面的布局等参数进行调整, 并且可以跳到子界面

a: 实现主界面跳转到子界面的代码如下

//跳转页面

```
public void toTmeCounter(View view) {  
    Intent intent = new Intent(this,TmeCounter.class);  
    startActivity(intent);  
}
```

(2) 子界面代码(activity_tme_counter.xml)

对子界面的布局等参数进行调整, 回到主界面可以直接右滑返回

(3) 采用的软件流程如下:

a: 判断波峰, 波峰出现前的过程, 加速度应当是逐渐增加的。

对两次连续采样得到的整体加速度 a 进行比较, 如果本次的 $a <$ 上次的 a , 则说明上次的 a 是加速度的波峰。为了避免走路时候的微小扰动影响判断, 设定一个精度范围 range , 当两次采样值的差的绝对值 $\geq \text{range}$ 以后才算作一次步行的峰值, 并将峰值的绝对值传给变量 a_{max} 。

b: 判断波谷, 波谷出现前的过程, 加速度应当是逐渐减少的。

对两次连续采样得到的整体加速度 a 进行比较, 如果本次的 $a >$ 上次的 a , 则说明上次的 a 是加速度的波谷。为了避免走路时候的微小扰动影响判断, 设定一个精度范围 range , 当两次采样值的差的绝对值 $\geq \text{range}$ 以后才算作一次步行的波谷, 并将波谷的绝对值传给变量 a_{min} 。

c: 取波峰和波谷绝对值的平均值 $0.5*(a_{\text{min}} + a_{\text{max}})$ 作为阈值, 每次得到波谷值以后, 如果波谷的绝对值大于该阈值则步数+1, 防止得到的波峰波谷值是微小扰动的波峰波谷值。

(4) 其计步过程的代码以及程序流程图如下:

a: 计步过程中的变量

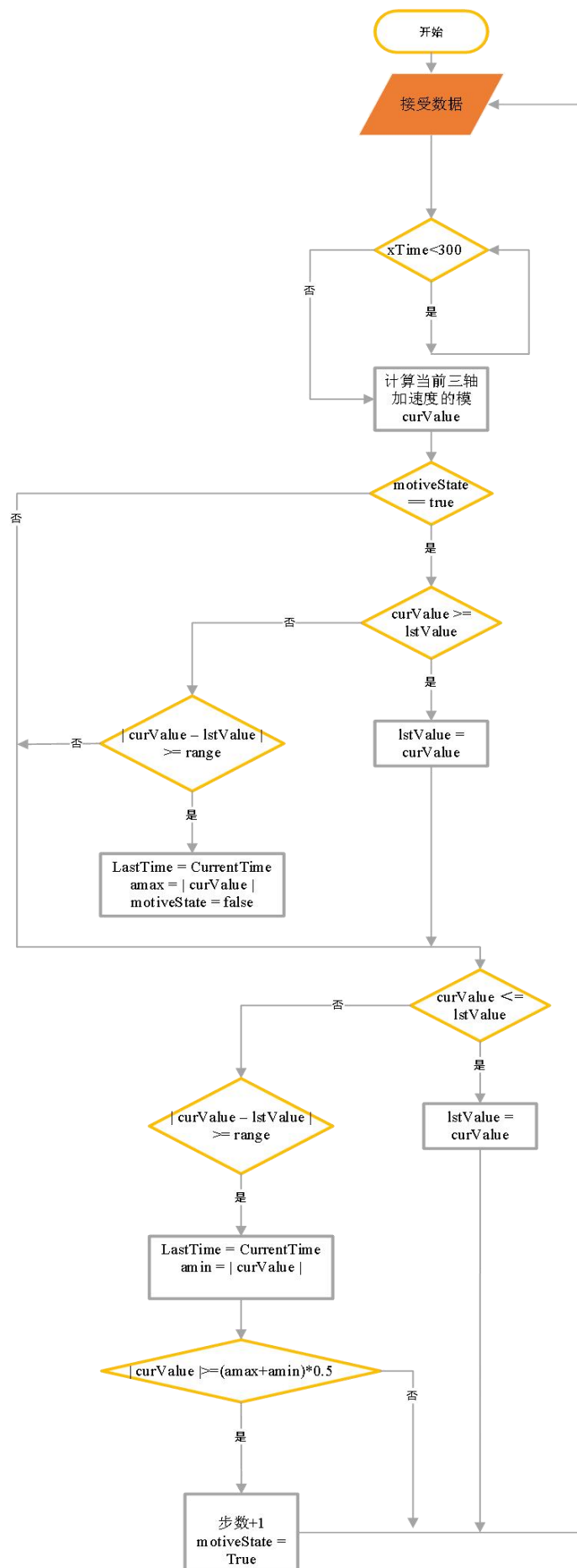
```
private double lstValue = 0; //上次的加速度值  
private double curValue = 0; //当前加速度值  
private boolean motiveState = true; //是否处于运动状态(默认处于运动状态)  
private double amax;//波峰绝对值  
private double amin;//波谷绝对值
```

```

double range = 25;//设定一个精度范围避免微小扰动
long xTime = CurrentTime-LastTime;//计步间隔
b: 计步的代码和流程图如下
float[] value = event.values;//取得 xyz 轴的加速度放入 value[0][1][2]
long CurrentTime = System.currentTimeMillis(); //取得当下时间:
long xTime = CurrentTime-LastTime;
if(xTime<300) return;//计步时间间隔不能太小
//计算当前的模
curValue=Math.sqrt(Math.pow(value[0],2)+Math.pow(value[1],2)+Math.pow(v
alue[2],2));
//取波峰和波谷的值
//向上加速的状态
if (motiveState == true) {
    if (curValue >= lstValue) lstValue = curValue;
    else {
        //检测到一次峰值
        if (Math.abs(curValue - lstValue) >= range) {
            LastTime = CurrentTime;
            amax = Math.abs(curValue);
            motiveState = false;
        }
    }
}
//向下加速的状态
if (motiveState == false) {
    if (curValue <= lstValue) lstValue = curValue;
    else {
        if (Math.abs(curValue - lstValue) >= range) {
            //检测到一次峰值
            amin = Math.abs(curValue);
            if(Math.abs(curValue)>=(amax+amin)*0.5){
                i++; //步数 + 1
                t0.setText(String.valueOf(i));//读数更新
                tv.setText("初加速度: \n"+x+", "+y+", "+z+"\n"+"末加速度:
\n"+DX+", "+DY+", "+DZ);
                t1.setText("触发间隔(ms): "+xTime);
                motiveState = true;
            }
        }
    }
}
}

```

c: 计步器的程序流程图如下



4 实验

将程序装入手机中，摇动手臂进行计步测试如下图：



打开倒计时模式后，输入 1000 秒倒计时然后进行计数，如下图：



可以看出实验基本满足了计步器的要求，然后在不同的条件下进行计步测试，取五次测量的平均值如表 1 所示：

表 1 步数测试表

计步方法	实际步数	测量步数
放在裤子的口袋中走路	50	38
握在手中走路	50	42

5 总结

总结本文的主要创新点,新发现,新问题,实验中存在问题和拟采取怎样的解决方案等。

5.1 本文计步方法的创新点

本文中的设计采取了动态阈值判断峰值的方法以及判断计步间隔的方法进行计步判断,还增加了一个精度值来判断得到的峰值是不是微小扰动产生的峰值。三种方法综合起来以提高计步的准确性。

其中动态阈值可以根据不同的步行状态(走路或者跑步)对阈值进行修改,计步间隔可以避免将快速的微小扰动时候的加速度变化判断为一步,精度值可以避免取得的峰值是源于微小扰。

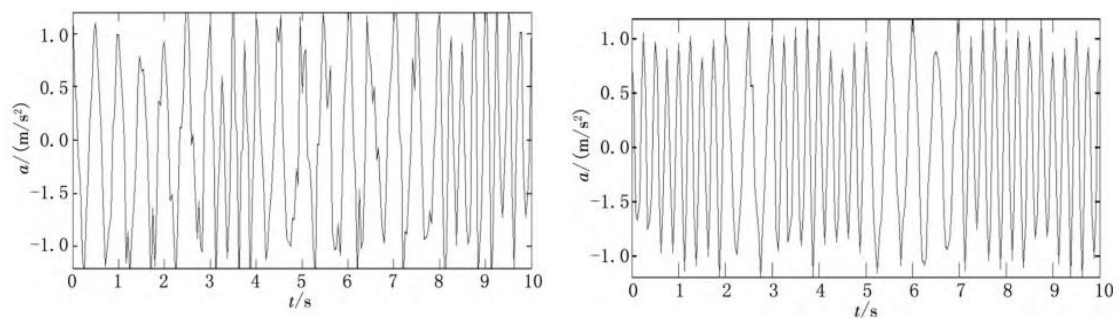
5.2 发现的问题

两个测试的步数均和实际的步数有较大差距,考虑应当是原始信号的随机噪声导致的。当随机噪声较多的时候,会导致多次峰值的出现间隔都在时间窗以外,从而将真正的峰值也判断为了扰动产生的峰值,步数就会比实际值要小。同时较多的随机噪声也会导致动态阈值的不准确,最终的计步就会出现误差。

5.3 拟采取的解决方案

由于本文虽然采取了一些措施来避免微小扰动的干扰,但是基本上还是把加速度的波形图当作比较理想的正弦图来考虑,实际的波形图是有很多的随机噪声,仅仅采取时间窗算法和动态阈值的自适应算法不能很好地避免噪声的影响,可以先采取滤波的方式来消除随机噪声,从而获得基本消除随机滤波的波形以后,再进行计步。

一种 LMS 自适应滤波消除随机噪声前后的步态信号如图所示^[6]。



滤波前原始步态信号

LMS 滤波后步态信号

图 2 LMS 自适应滤波前后的步态信号

参考文献

- [1] 蔚利娜. 基于加速度的计步算法和步长计算研究与实现 [D]; 东北大学, 2013.
- [2] 陈国良, 张言哲, 杨洲. 一种基于手机传感器自相关分析的计步器实现方法 [J]. 中国惯性技术学报, 2014, 22(06): 794-8.
- [3] 谢如花. 步数检测方法及在手腕式计步器中的应用研究 [D]; 兰州交通大学, 2013.
- [4] 徐思远. 基于手机传感器的计步方法研究与系统实现 [D]; 江苏大学, 2017.
- [5] 赵琦 苏 A 董 A. 基于加速度传感器的计步器系统 [Z]. 全国第二届嵌入式技术联合学术

会议论文集. 黑龙江牡丹江. 2007: 163-5

[6] 梁潘 晏 A 雷 A 周 A. 基于三轴加速度传感器的自适应计步器的实现 [J]. 东北师大学报 (自然科学版) %J Journal of Northeast Normal University(Natural Science Edition), 2016, 48(3): 79-83.

附录

a: 为了使用定时器在子线程中更新 ui 线程, 采取了 handler 开启线程刷新进行处理, 其倒计时界面的开始计时按钮槽函数内容如下所示

```
starttime.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //步数清 0,重头开始计数
        i = 0;
        t0.setText(String.valueOf(i));
        //开启线程刷新,从而在子线程中更新 ui 线程,实现倒计时
        handler = new Handler();
        timer = new Timer();
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        if(j>0)
                        {
                            j--;
                            time.setText(String.valueOf(j));
                        }
                    }
                });
            }
        },1000,1000);
    }
});
```

b: 为关闭倒计时的代码如下所示

```
stoptime.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        timer.cancel();//关闭定时器
        handler.removeCallbacksAndMessages(null);
    }
});
```