

多线程模拟电梯运行算法

潘 昊, 高文杰

(武汉理工大学 计算机科学与技术学院, 湖北 武汉 430070)

摘 要: 电梯的运行模拟, 是计算机控制教学中的重要素材。模拟方法主要分为硬件模拟和软件模拟, 硬件模拟是以单片机为核心, 辅以相关硬件和控制程序来实现; 软件模拟则是以软件全程模拟电梯系统, 包括用户界面和电梯的控制。文中主要探讨了在 VC 环境下, 实现软件模拟电梯运行的算法设计和程序实现, 实验证明, 堆栈和多线程技术是软件模拟电梯运行的有力工具。

关键词: 模拟电梯运行; 堆栈; 多线程

中图分类号: TP391.9

文献标识码: A

文章编号: 1673-629X(2006)08-0155-02

Algorithm on Simulating Elevator Running with Multithreading

PAN Hao, GAO Wen-jie

(College of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract: The simulation of elevator running is an important teaching material in computer control, which is classified as hardware simulation and software simulation. Hardware simulation is realized with SCM, assistant hardware and control program. Moreover, software simulation means that simulate the whole elevator system including UI and elevator control in pure software environment. This article mainly discussed the algorithm design and the method of programming on simulating elevator running, using stack as data structure, VC as developing platform. Experimental result indicates that stack and multithreading are effectual in software simulation of elevator running.

Key words: simulating elevator running; stack; multithreading

0 引言

软件模拟电梯运行算法有很多, 这些算法大多采用以结构体为元素的数组作为数据结构。通过对电梯运行规律的分析, 可以发现电梯总是优先响应处于电梯“前方”且距离最近的同向请求, 据此文中采用了另外一种数据结构——堆栈。堆栈中存放电梯的楼层数, 栈中元素由栈顶至底距离电梯由小到大依次排列, 电梯持续读取栈顶元素, 则可以完成电梯运行的模拟。

1 电梯的运行规律

电梯接收的请求分为两大类:

(1) 内部请求: 用户在电梯内部发出的请求, 显然, 内部请求都是处于电梯“前方”的同向请求。例如用户在电梯内部按下 12 楼的按钮。

(2) 外部请求: 用户在电梯外发出的请求。例如 12 楼的用户按下了“向下”的按钮。外部请求又分为 3 种:

- ① 处于电梯“前方”的同向外部请求;
- ② 反向外部请求;
- ③ 处于电梯“后方”的同向外部请求。

从效率考虑, 电梯应该沿着当前方向前进尽可能远的距离然后再反向。因此, 当电梯运行时, 对沿途同向的外部请求做出响应; 对反向和处于电梯“后方”的外部请求暂时不做响应(但应做记录以便将来响应), 即电梯除了响应内部请求外, 还要响应外部沿途同向的请求。当电梯“前方”没有同向的请求, 则此时要判断“前方”是否有反向的请求, 若有反向请求, 则应响应“最前方”的反向请求。即电梯此时应到达“最前方”的楼层, 然后转向; 若没有反向请求, 则电梯应该转向^[1]。

2 算法设计

2.1 数据结构的设计

通过对用户请求种类的分析可以发现, 对电梯而言, 内部请求与处于电梯“前方”的同向外部请求没有区别, 据此, 可以设置 3 个堆栈来存储这 4 类请求。

- 1) 存储内部请求和处于电梯“前方”的同向外部请求的堆栈, 命名为 SCP;
- 2) 存储反向外部请求的堆栈, 命名为 SCS;
- 3) 存储处于电梯“后方”的同向外部请求的堆栈, 命名为 SA。

因此, 本算法使用的堆栈的入栈函数必须做出相应改动, 以实现上述功能^[2]。

收稿日期: 2005-11-13

作者简介: 潘 昊(1964-), 男, 湖北孝感人, 博士, 副教授, 研究方向为神经网络。

2.2 算法设计

1) 用户请求的存储策略。

如上所述,不同的用户请求将存储在与之对应的堆栈中。同时,堆栈中元素的顺序受到电梯方向的影响,当电梯向上时,SCP 和 SCS 栈中元素由栈顶至底升序排列,SA 栈中元素由栈顶至底降序排列;反之,当电梯向下时,SCP 和 SCS 栈中元素由栈顶至底降序排列,SA 栈中元素由栈顶至底升序排列。

2) 电梯对用户请求的响应机制。

电梯从当前楼层出发,不断读取 SCP 栈顶元素作为目的楼层,到达目的楼层后,停靠,再把该楼层作为当前楼层,如此循环,直到电梯转向;当电梯转向时,相应地,3 个堆栈将发生相互转化,SA 转化为 SCP,SCP 转化为 SCS,SCS 转化为 SA;此后,电梯再次开始循环读取 SCP 栈顶元素,如此循环,完整地模拟出电梯的运行(如图 1 所示)。

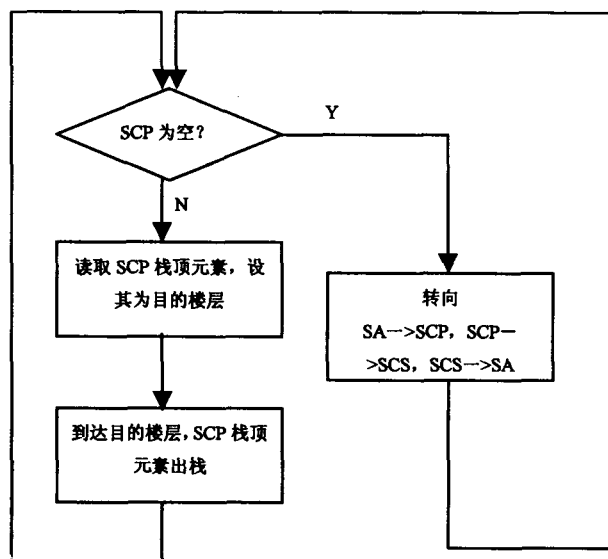


图 1 模拟电梯运行程序流程图

3 程序实现

本算法在 VC 环境下使用多线程作为程序实现。

电梯系统的运行分为两个方面:

- 1) 用户随时按下按钮发出请求;
- 2) 电梯运行,适当地响应用户请求。

因此,采用多线程编程作为程序实现是比较合理的,利用主线程负责 UI,利用辅助线程模拟电梯运行^[3]。

初始状态下,当用户发出请求时,电梯线程被开启(若电梯线程已经开始运行时,用户发出请求不会开启一个电梯线程)。

为确定电梯转向时机,设置一个全局变量 int vertex; 以标识电梯转向时所处位置,在本算法中称之为“顶点”。

功能实现的核心过程:

1) 用户请求的发出与存储。

以按钮控件作为请求发出装置,按下按钮,发出请求信息(内部请求为目的楼层;外部请求为所处楼层)并存入

相应的堆栈。

以 15 楼用户发出向上的请求为例:

```
void CLiftDlg::OnUpBtm15()
```

```
{
    更新顶点;
    若电梯未启动则启动电梯;
    发出请求;
}
```

2) 电梯线程:

```
UINT Run(LPVOID pParam)
```

```
{
    等待被唤醒;
    while (1)
    {
        while (SCP 栈非空)
        {
            开往目的楼层;
            到达目的楼层,栈顶元素出栈;
            //判断停靠时机
            if (停靠条件满足)
                到达目的楼层,停靠十秒钟;
        }
        //SCP 栈为空,判断电梯变向时机
        if (SCP 栈为空)
        {
            //三栈全空,电梯停止运行
            if (SCS 栈和 SA 栈都为空)
                return 0;
            //未到达顶点,不能变向
            if (未到达顶点)
                顶点压入 SCP 栈;
            if (到达顶点)
            {
                //有反向请求
                if (SA 栈非空)
                {
                    SA->SCP, SCP->SCS, SCS->SA;
                    电梯转向;
                }
                if (SA 栈非空)
                    以 SA 栈顶元素更新顶点;
            }
            /* 没有反向请求,但有出于电梯“后方”的同向请求
            * /
            else if (SCS 栈非空)
            {
                SCS 栈顶元素压入 SA;
                SA->SCP, SCP->SCS, SCS->SA;
                电梯转向;
            }
        }
    }
}
```

(下转第 159 页)

进行左手运动想象的时候, C3 电极的 mu 节律能量明显大于 C4 电极的 mu 节律能量;而对于右手想象运动则正好相反。并且这个规律在整个分析区域内存在,因此有利于 BCI 系统对不同运动想象的分类和决策。

根据上图,建立一简单的基于 mu 节律能量为特征的左右手运动想象分类方法:

$$m_{2C3}(t) - m_{2C4}(t) \begin{cases} > 0 & \text{左手运动想象} \\ < 0 & \text{右手运动想象} \end{cases} \quad (5)$$

式中 t 为决策分类时间。为了验证上述结果是否具有普遍的普遍性,根据这一分类判决准则,对 140 组带运动想象类型标注的 EEG(C3, C4) 信号进行动态分析。取决策时间 $t = 7.4219$, 窗口长度 $N = 512$, 得到 EEG(C3, C4) 信号基于 mu 节律能量的决策分类图如图 3 所示。

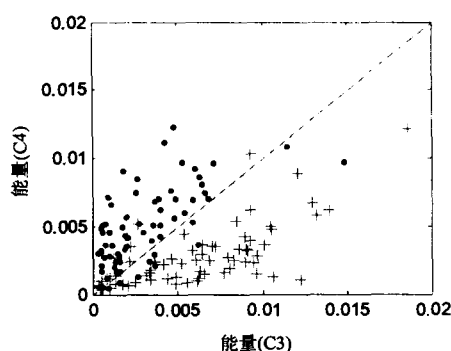


图 3 基于 mu 节律能量的 140 组脑电信号的决策分类图 (“+”:左手,“.”:右手)

经计算分类正确率可达到 85%。值得一提的是,笔者对那些用本方法分类错误的实验数据进行了认真分析,发现识别错误的主要原因是这些数据的 ERS(ERD)特征不明显,这与被测试者在脑电数据采集时的精神状态有关(如精神不集中、疲劳等),对于这类脑电数据,用其他识别方法的效果也不好。

(上接第 156 页)

4 结束语

文中通过对许多电梯模拟算法的学习,结合自己对电梯运行规律的认识,提出了一种以堆栈作为数据结构、以多线程作为程序实现的电梯模拟算法。

相对于数组,堆栈更能真实地反映电梯响应用户请求的规律,即电梯总是优先响应处于电梯“前方”且距离最近的同向请求,使用多线程也比较合理地反映了电梯系统中、电梯与用户之间相对独立的关系。

本算法有几个关键点:

- 1) 用户请求的区分以及区别性地存储;
- 2) 电梯转向时机的确定;

3 结束语

实验表明,基于 mu 节律能量的运动意识的分类方法能够从记录的脑电时间序列中及时识别与运动想象相关的脑电模式,若将这些特征转化为某种控制信号处理控制光标的移动或辅助运动设备,就可以帮助严重瘫痪病人通过计算机与外界环境进行交流和沟通或者帮助他们实现自己想做的动作。基于 mu 节律能量的分类方法与其他分类方法相比最大优点在于它计算简单,而且可在线计算,故可认为,基于 mu 节律为脑电特征的意识任务分类在脑机接口的应用中有较高的实用价值。

参考文献:

- [1] Wolpaw J R, Birbaumer N, McFarland D J. Brain - computer interfaces for communication and control[J]. Clinical Neurophysiology, 2002, 113: 767 - 791.
- [2] Hinterberger T, Kübler A. A brain - computer interface (BCI) for the locked - in: comparison of different EEG classifications for the thought translation device[J]. Clinical Neurophysiology, 2003, 114: 416 - 425.
- [3] Babiloni F, Cincotti F, Bianchi L. Recognition of imagined hand movements with low resolution: surface Laplacian and linear classifiers[J]. Medical Engineering & Physics, 2001, 23: 323 - 328.
- [4] Costa E J X, Cabral Jr E F. EEG - based discrimination between imagination of left and right hand movements using adaptive gaussian representation[J]. Medical Engineering & Physics, 2000, 22: 345 - 348.
- [5] Neuper C, Müller G R, Kübler A. Clinical application of an EEG - based brain - computer interface: a case study in a patient with severe motor impairment[J]. Clinical Neurophysiology, 2003, 114: 399 - 409.

3) 电梯转向时堆栈之间的转化(利用指针实现)。

完整的电梯模拟算法还涉及到电梯运行时所处楼层的动态显示、停靠时的减速、对负载的限制等细节问题,这些问题的解决办法较为固定,文中未做叙述,有兴趣的读者可以参考文献[4, 5]。

参考文献:

- [1] 曹荣敏. 电梯运行系统的单片机模拟[J]. 微处理机, 1998 (3): 1 - 3.
- [2] 严蔚敏. 数据结构[M]. 北京: 清华大学出版社, 1997.
- [3] Huges C. C++ 面向对象多线程编程[M]. 周忠良译. 北京: 人民邮电出版社, 2003.
- [4] 金 沙. 模拟电梯运行的控制系统[J]. 渤海大学学报, 2004 (2): 1 - 3.
- [5] 曹荣敏, 刘满仓, 谢 宇. 高层建筑中电梯运行系统的模拟[J]. 计算机应用研究, 1999(2): 1 - 2.