

# 智能跑步机上虚拟场景的设计与实现

李春丽<sup>1,2</sup> 龚少兵<sup>1</sup> 杨先军<sup>1,2</sup> 姚志明<sup>1</sup> 周旭<sup>1</sup> 孙怡宁<sup>1</sup>

(1. 中国科学院合肥智能机械研究所, 安徽合肥 230031; 2. 中国科学技术大学, 安徽合肥 230027)

**摘要:** 为解决传统跑步机锻炼的单调乏味性, 文章应用虚拟现实技术, 通过三维建模, 基于 Visual Studio 和 OpenGL 平台开发实现了应用于智能跑步机上的虚拟场景, 增添跑步机上锻炼的户外临场感和趣味性。通过实验测试, 场景运行良好。

**关键词:** 跑步机; 虚拟场景; 三维建模; OpenGL; Visual Studio

中图分类号: TP391

文献标识码: A

文章编号: 1006-2394(2012)09-0046-03

## Design and Implementation of an Intelligent Treadmill Based on Virtual Scene

LI Chun-li<sup>1,2</sup>, DOU Shao-bin<sup>1</sup>, YANG Xian-jun<sup>1,2</sup>, YAO Zhi-ming<sup>1</sup>, ZHOU Xu<sup>1</sup>, SUN Yi-ning<sup>1</sup>

(1. Institute of Intelligence Machines, Chinese Academy of Sciences, Hefei 230031, China;

2. University of Science and Technology of China, Hefei 230027, China)

**Abstract:** In this paper, to reduce the monotony of traditional treadmills, the Virtual Reality technology is utilized to develop the virtual scene applied to intelligent treadmills via 3D modeling. All the development is based on Visual Studio and OpenGL. With the add-on of the virtual scene, the treadmill is endowed with more playfulness and the telepresence of outdoor exercise. Experiments demonstrate that the virtual scene on the treadmill works well.

**Key words:** treadmill; virtual scene; 3D modeling; OpenGL; Visual Studio

## 0 引言

本文将虚拟现实技术融入到传统的跑步机中, 使锻炼者可以融入到虚拟环境中, 增强户外运动的真实感, 实现人与虚拟环境的直观自然地交互。

## 1 虚拟现实技术简介

所谓的虚拟现实(Virtual Reality, VR)技术, 是指利用计算机模拟的人工环境代替现实世界的真实自然环境, 增添户外实地接触感觉和贴近自然的场景呈现, 给人身临其境的感觉, 它综合利用计算机图形图像技术、仿真技术、多媒体技术、人工智能技术和多传感器技术等, 来模拟人类感觉世界的实时表示, 这里的“世界”既可以是某种特定现实环境的再现, 也可以是纯粹构想的虚拟场景。VR系统具有多种输出形式和处理多种输入设备的能力, 能够进行碰撞检测、实时交互、视点控制和复杂行为建模等。

## 2 虚拟场景的实现

在基于虚拟现实技术的跑步机系统中, 建立模型和实时绘制是最基本的技术, 在模型的精细程度

和绘制速度方面, 既要保证一定的显示质量, 又不能造成用户运动时的不适感。场景的建模需要用到3D建模软件, 常用的3D建模软件有: 3D max、Maya及MilkShape 3D等, 其中MilkShape 3D是一款比较简单实用的软件, 本文中采用这款软件来设计所需要的场景。

### 2.1 场景建模

虚拟场景建模主要包括以下几个步骤:

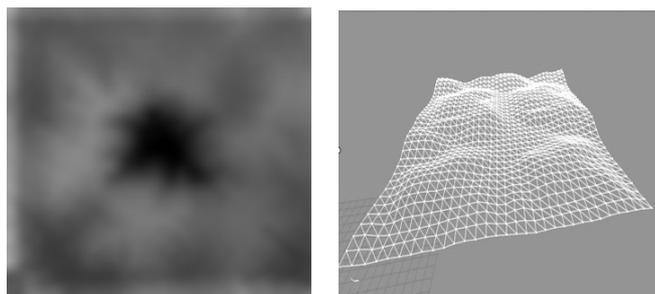
(1) 绘制高度图: 高度图是记录高度信息的图片, 它的单元格式可以是任意的, 一般情况下, 使用单字节或双字节作为一个单元的大小。常用raw格式的文件保存8位高度信息, 数据限制在[0 255]区间, 其中0是最低点, 255是最高点。

(2) 然后利用辅助程序直接将高度信息从文件读取到内存, 转换成MilkShape 3D软件可读的raw(terrain.raw)格式, 而不是通过计算随机生成。使用MilkShape 3D导入terrain.raw, 生成三角形组成的地形。地形的生成如图1所示。

(3) 绘制平面跑道: 基于三角形单元, 使用MilkShape 3D建立跑道的平面模型, 并导出.raw格式文件(temptrack.raw)。图2所示为跑道的平面模型。

收稿日期: 2012-04

作者简介: 李春丽(1986—), 女, 硕士研究生, 研究方向为智能运动装备。



(a) 高度图 (b) 三角形地形

图 1 地形的生成

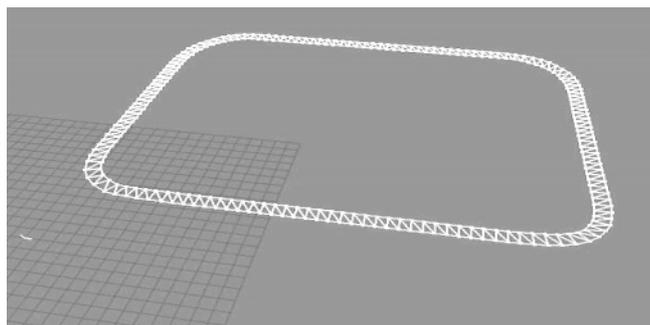


图 2 平面跑道

(4) 跑道和地形的相对位置调整: 使用 Milk-Shape 3D 同时导入 temprack. raw 和 terrain. raw ,调整垂直的相对位置。利用辅助程序 ,使用 terrain. raw ,调整 temprack. raw ,生成随地形起伏的跑道文件 track. raw 。

(5) 天空的生成: 天空的生成借助于天空盒的方法。所谓天空盒是指一个长方体的盒子 ,6 个面都贴上对应的天空纹理。在应用过程中 ,只生成一个 5 个面的盒子 ,底面是地面。变量 m\_texture 用来保存导入的 5 张天空纹理。在选择天空的贴图纹理时 ,需要注意图片是否首尾相连 ,避免出现缝隙。在程序初始化函数里面添加载入天空纹理的图片。导入图片之后 ,采用函数 CreateSkyBox 来建立天空盒 ,函数 CreateSkyBox 的部分代码如下:

```
void object::CreateSkyBox( int a jint wi jint he jint le) //显示天空
{
    float width = MAP* wi; //天空盒宽
    float height = MAP* he; //天空盒高
    float length = MAP* le; //天空盒长
    float x = MAP - width/2; //天空的位置 x
    float y = MAP/a - height/2; //天空的位置 y
    float z = -MAP - length/2; //天空的位置 z
    texture( m_texture[2] ); //设置 BACK 贴图左
    glBegin( GL_QUADS ); //多组独立填充四边形
        glVertex2f( 1.0f, 0.0f ); glVertex3f( x + width, y, z );
        glVertex2f( 1.0f, 1.0f ); glVertex3f( x + width, y + height, z );
        glVertex2f( 0.0f, 1.0f ); glVertex3f( x, y + height, z );
        glVertex2f( 0.0f, 0.0f ); glVertex3f( x, y, z );
    glEnd();
}
```

```
.....
..... }
```

## 2.2 路径规划

由于在跑步机上难以自由地控制方向 ,所以 ,需要对场景进行“路径规划” ,以便场景可以自动沿着既定的路线向前移动。对于使用不规则三角网模型 ( Triangulated Irregular Networks , TIN) 方式绘制的跑道 ,在规划路径时可以有很多种方案 ,如选择三角形的中心、垂心、形心等。但是在实际应用中发现 ,由于三角形位置的关系 ,这样的方法规划出来的路径都是曲折的 ,效果不佳。为此本文设计出沿着三角形一边的中点行进的路径规划方案 ,这种方案规划出来的路径平滑度较好。在程序设计时 ,需要实时根据跑道的位置来找到对应三角形的相应边进行计算 ,得到前移的路径。

## 2.3 曲线平滑

由于规划出来的路径是离散的点表示的 ,如果按照这些点连接的折线前进 ,在三个维度的方向上都会出现跳动的现象。这就要找到一条平滑的曲线 ,即找到一个高效的曲线拟合算法 ,要求速度快 ,以满足实时要求 ,同时还要求精度可变。对于此类曲线拟合的问题 ,常用的算法是采用双线性插值以及最小二乘法等。

在实际应用中发现 ,这些方法拟合的效果虽好 ,但是实现复杂 ,计算量大 ,难以满足实时的要求。为此 ,本文设计出一种比较快速的曲线平滑的方法( 图 3) ,即在三个维度上 ,均让视角的变换沿着内部插入点形成的圆的切线方向进行 ,从而消除切换抖动。

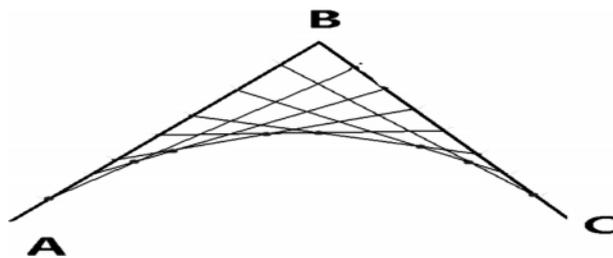


图 3 曲线平滑

## 2.4 消息映射

为实现在运动的过程中对跑步机进行控制 ,需要在场景之上添加一些控制按钮 ,这些按钮主要有加速、减速、音乐、媒体和退出等。绘制菜单背景框 ,其中菜单背景框为蓝色 ,使用 RGB 混合形成半透明效果 ,然后在菜单框的适当位置绘制按钮。由于 OpenGL 的渲染管线是单向的 ,难以从屏幕二维图像回到三维世界中 ,也就是说 ,不能通过使用鼠标在屏幕中的位置来进行按钮的响应操作。为此 ,使用 OpenGL 中的拾取技术 ,识别不同的按钮对象。

在 OpenGL 中 ,提供响应函数: gluPickMatrix ( x ,

$y$ ,  $width$ ,  $height$ ,  $viewport$  [4]); 其中  $x$ ,  $y$  是鼠标点击到窗口上的坐标,  $width$  和  $height$  就是这个拾取框的长和宽,  $viewport$  是窗口的大小。调用该函数之后, OpenGL 就会创建一个拾取矩阵, 分解这个矩阵, 可以看到, 这个矩阵就是上面的移动拾取框到原点, 然后再放大为视体大小。

### 3 效果检验

虚拟场景制作完成之后, 需要提取作用于跑步机产生的各种运动参数(速度、里程、心率等), 输入到计算机实时显示, 作用于虚拟场景, 实现健身者与虚拟环境的实时交互。由于 Windows 窗口的设备上下文被 OpenGL 的渲染环境所替换, 无法使用  $TextOut()$ 、 $DrawText()$  等函数进行文字显示, 所以采取创建字体, 然后在内存中生成字体位图进行显示的方法。图 4 为实时运行效果图。



图 4 虚拟场景实时运行图

### 4 结论与展望

虚拟现实技术与体育健身器材的完美结合正在逐步改变人们的生活方式, 基于虚拟现实技术的智能跑步机, 必将改变大众的健身方式。本文将虚拟现实技术应用于跑步机上, 增添锻炼时的趣味性和真实自然场景的呈现, 能够帮助健身者坚持执行运动健身计划。本文中的虚拟场景只是原理上的简单实现, 今后还有很多工作需要完善, 比如, 可以实现在虚拟场景中添加一个人物角色、获取更多丰富的参数作用于人物角色、丰富场景等。

#### 参考文献:

[1] 钱克宠, 胡维华. 虚拟现实技术在跑步机中的设计与实现[J]. 计算机仿真, 2009, 26(5): 259-261.

[2] 王永强. 基于虚拟现实技术的多功能跑步机系统的研究与设计[J]. 科技信息, 2010(5): 449-450.  
 [3] 王鸿灏. 计算机网络虚拟实验系统的设计与实现[D]. 北京: 北京邮电大学, 2008.  
 [4] 纪庆革, 潘志庚, 李祥晨. 虚拟现实在体育仿真中的应用综述[J]. 计算机辅助设计与图形学学报, 2003, 15(11): 1333-1338.  
 [5] Dave Shreiner. OpenGL 编程指南[M]. 7 版. 北京: 机械工业出版社, 2010. (郁菁编发)

(上接第 23 页)

表 1 实车横向控制精度

速度/( $m \cdot s^{-1}$ )	0~2	2~4	4~6	6~8	8~10
最大横向误差/m	0.29	0.25	0.27	0.24	0.20
最小横向误差/m	0.11	0.17	0.15	0.12	0.13
横向控制平均误差/m	0.25	0.23	0.20	0.24	0.18

表 2 实车纵向控制精度

速度/( $m \cdot s^{-1}$ )	0~2	2~4	4~6	6~8	8~10
预定车速平均值/( $m \cdot s^{-1}$ )	1.65	2.89	4.55	7.04	8.76
实际车速平均值/( $m \cdot s^{-1}$ )	1.33	3.22	4.98	6.56	9.23
纵向控制平均误差/( $m \cdot s^{-1}$ )	0.32	0.33	0.43	0.48	0.47

实验结果表明, 在各种速度情况下, 汽车在路径跟踪时的横向控制误差在 0.3 m 以内; 纵向控制误差在 0.5 m/s 以内。实验具有良好的重复性, 体现该算法在实际应用中的适用性。

### 4 结语

通过对路径跟踪算法的研究, 设计了适合嵌入式系统的汽车横向控制和纵向控制模型, 最后通过实车实验的数据分析证明了算法的有效性。基于路径跟踪算法的嵌入式控制系统已经在实车平台正常运行 4 个多月, 实验测试里程超过 2000 km, 能够完成 50 km/h 平均时速下的预定轨迹的自动驾驶, 有效地验证了该算法的稳定性。

#### 参考文献:

[1] 张鹏飞, 何克忠. 多功能室外智能移动机器人实验平台[J]. 机器人, 2002, 24(2): 97-101.  
 [2] Broggi A, Bertozzi M, Fascioli A. ARGO and the MilleMiglia in Automatico Tour[J]. Proceedings of the IEEE Intelligent Systems, 1995, 14(1): 55-64.  
 [3] 谯艳娟. 汽车纵向动力学系统控制方法的研究[D]. 重庆: 重庆大学, 2005.  
 [4] Jean J. Labrosse. 嵌入式实时操作系统  $\mu C/OS-II$  [M]. 2 版. 邵贝贝, 译. 北京: 北京航空航天大学出版社, 2003.

(许雪军编发)