

文章编号: 1009-671X(2004)03-0030-03

基于 OpenGL 的助降灯光界面系统的开发

朱胜缘, 杨东梅, 朱齐丹

(哈尔滨工程大学自动化学院, 黑龙江哈尔滨 150001)

摘 要:提出了基于 OpenGL 的助降灯光控制系统中界面系统的开发方法. 详细介绍怎样通过 OpenGL 来实现灯光模拟技术的设计, 其中涉及了视景体的选择、光照、纹理等等的处理. 实践证明该仿真系统具有一定的可观性和交互友好性.

关 键 词: OpenGL; 光照; 纹理

中图分类号: TP39.9 **文献标识码:** A

Development of a controlling system of landing lighting based on OpenGL

ZHU Sheng-yuan, YANG Dong-mei, ZHU Qi-dan

(School of Automation, Harbin Engineering University, Harbin 150001, China)

Abstract: A developing method was presented for an interface controlling system of landing lighting based on OpenGL. The design and realization of light simulation were introduced in detail, including how to deal with the model, illumination and texture etc. The system has its important value for users.

Key words: OpenGL; illumination; texture

助降灯光系统是保障飞机在夜间和复杂天气条件下在船舶上进行顺利起飞、着陆、滑行的目视助降设备. 助降灯光系统的工作状况、可靠性、应急性与飞机的安全紧密相关.

以前, 助降灯光需要工作人员亲自到现场进行监测, 对相应设备进行手动操作并记录操作过程. 基于 OpenGL 的助降灯光界面系统应用后, 工作人员可以根据能见度情况直接对灯光系统监测并进行及时的控制和调整, 避免了直接到现场这一环节, 提高了操作可靠性. 灯光维护人员可以在室内通过对显示界面的观察并根据报警信息来及时地掌握灯光系统的实时运行状况并排除故障, 有效地消除了故障积累的现象, 同时又减少了工作量. 而且该仿真系统具有一定的可观性, 增强了使用人员与系统之间的交互友好性.

1 系统模块的设计

系统的功能主要由以下三大模块组成.

1.1 视景体模型的设置

如果对 OpenGL 的有关视景体设置的投影变换函数设置不当, 图形不能出现在视口或只有部分图形出现. OpenGL 提供了 2 种投影方式: 平行投影和透视投影, 2 种投影的使用方法是类似的. 平行投影又叫正交投影, 它的视景体为一个长方体, 由其产生的投影不会因物体的远近而改变大小尺寸(见图 1); 透视投影的视景体为一个台锥, 这样使远视点的物体小, 近视点的物体大, 更符合三维模拟仿真环境, 以透视投影为例, 它的视图体是一个圆棱锥的平截台体, 即是一个被平行于底面的平面截掉顶端的锥棱(见图 2). 同样一个物体, 距离视点较近将比远离视点时在视图体中占据更大的比例, 因此在平截台体的粗端将显得大一些. 由于这种投影方法类似于人眼的视觉机制, 所以通常用于强调真实感的场合. 本文主要采用透视投影, 其设置由投影变换函数 `glFrustum` 来实现. 在图形生成过程中, 有关视景体的处理步骤

为:首先设定一个视景物,把物体移到视景物当中,视景物以外的物体都被裁剪掉.再进行投影变换,绘制各个物体.

1.2 接收信息和处理信息模块

系统采用 PC 机作为上位计算机对直升机灯光助降控制系统及模拟设备进行管理,该上位管理计算机同时进行 1 路视频监控,具有视频切换、方位控制及调焦功能.其中上位机和下位机信息的通讯是通过 RS-485 总线及 CAN 总线网络形式来实现的.上位机通过插在 PCI 总线插槽上的接口卡接入 CAN 总线,下位机通过嵌入的通讯功能模块接入 CAN 总线.

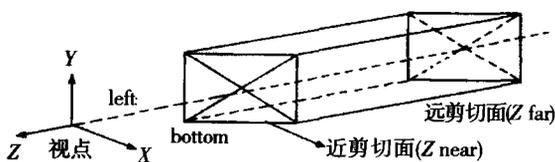


图1 平行投影变换图

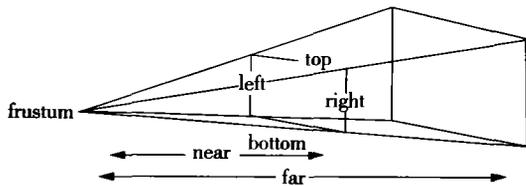


图2 由 glFrustum() 定义底透视投影变换视图体

接口卡完成的功能有:

1) 数据转发的功能. CAN 接口卡收集 CAN 总线上各个节点的信息,转发给 PC 机,并将 PC 机的命令和数据转发给各个节点.

2) 监控、管理功能.完成对 CAN 总线上的用户系统的部分监控和管理工作的.

整个系统接收的数据信息主要有:系统中的灯状态,包括灯是否短路,是否为正常工作状态,是否为故障等,以及报警器浓度值.为了保证所有的数据接收无误且不丢失,制定了相应的接口协议.在本文中,上位机接收到下位机的数据信息为二进制信息,应根据通讯协议经过函数处理转化成所需的实际信息,并反映到界面上加以显示.

1.3 灯光效果实现模块

在真实感图形学中,为了模拟现实世界中的场景,需要知道这个场景的光照效果的物理模型,然后用一个数学模型来表示它,通过计算数学模型可以得到计算机模拟出来的真实感效果.在现

实世界中,光照效果一般包括光的反射、光的透射、表面纹理和阴影等.

1.3.1 定义光源

光源函数为^[1] void glLight (GLenum light , GLenum pname , TYPE param), glLight 建立 light 指定的光源. light 用形式为 GL_LIGHTi 的符号常数表示,本文中设定为 GL_LIGHT0,即选用光源 0,同时将它定义成白色光源作为整个环境的光源.定义完毕后,用 glEnable(GL_LIGHT0) 打开该光源.

```
GLfloat light0_ambient[4] = {0.5f, 0.5f, 0.5f, 1.0f};
```

```
GLfloat
```

```
light0_diffuse[4] = {1.0f, 1.0f, 1.0f, 1.0f}; GLfloat
```

```
light0_specular[4] = {1.0f, 1.0f, 1.0f, 1.0f};
```

```
GLfloat light0_position[4] = {100.0f, 0.0f, 1000.0f, 0.0f};
```

```
glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse);
```

1.3.2 为场景定义光照模式

OpenGL 的光照模式由 3 个部分组成:全局环境光的强度,观察点的性质,对物体的前、背面的光照处理方式.本文中通过以下语句来设置的:

```
GLfloat model0_ambient[] = {0.4, 0.4, 0.4, 1.0};
```

```
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel0_ambient);
```

1.3.3 设定材质属性

大多数的材质属性与光源的属性类似,材质属性的设置对物体的显示效果影响很大,尤其在本文中,要真实地模拟灯光,需重点考虑材质.例如,要绘制红灯,定义其材质如下:

```
void CMyView::AddRedMaterial()
```

```
{
```

```
GLfloat no_mat[] = {0.0, 0.0, 0.0, 1.0};
```

```
GLfloat mat_Red shininess[] = {3.0};
```

```
GLfloat mat_Red_diffuse[] = {0.2f, 0.2f, 0.2f, 1.0}; GLfloat mat_emission_Red[] = {1.0f, 0.0f, 0.0f, 0.0};
```

```
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_Red_diffuse);
```

```
glMaterialfv(GL_FRONT, GL_EMISSION, mat_emission_Red);
```

```

}
    设置 BOOL 型变量, 结合定时器和 if 选择语
    句来实现灯光闪烁的效果.

```

1.4 纹理的实现

例如要实现天空的再现, 主要通过以下函数的编写来实现:

```

BOOL CMyView::LoadSkyImageData()
{
    HBITMAP hBmp = ( HBITMAP ):: LoadImage
    ( NULL, " sky. bmp", IMAGE_BITMAP, 0, 0, LR_
    LOADFROMFILE | LR_CREATEDIBSECTION );
    BITMAP BM_SKY; ::GetObject( hBmp, sizeof( BM_
    SKY ), &BM_SKY );
    long lSize = BM_SKY. bmWidth * BM_SKY. bm_
    Height;
    m_lSkyWidth = BM_SKY. bmWidth;
    m_lSkyHeight = BM_SKY. bmHeight;
    BYTE * pBM = ( BYTE * ) BM_SKY. bmBits;
    m_bpSky = new BYTE[ lSize * 4 ];
    for( long i = 0; i < m_lSkyHeight; i + + )
    { ..... }
    return TRUE;
}

void CMyView::SetSkyTexture()
{
    glPixelStorei( GL_UNPACK_ALIGNMENT, 1 );
    glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE
    _MAG_FILTER, GL_LINEAR );
    glTexImage2D( GL_TEXTURE_2D, 0, 4, m_lSky_
    Width, m_lSkyHeight, 0, GL_BGRA_EXT, GL_UN_
    SIGNED_BYTE, m_bpSky );
}

```

2 人机交互界面设计

整个系统由四大部分组成: 灯光系统、报警器系统、模拟设备系统和图像采集. 为了生动形象地展现整个系统的状态, 采用全屏、多窗口切换分层显示技术. 前 3 个系统都有主界面、动画显示、面板显示、表格显示、打印 5 项功能, 其显示可通过键盘任意切换. 主界面模拟了船舶在大海中的运

动姿态, 其中船舶航行中的大海, 天空都是采用 OpenGL 特有技术如: 纹理贴图技术、移动纹理及物体运动模糊效果雾化效果来实现的. 而且通过相应控制能移动船体, 并进行旋转以便从不同的方向观察船体. 动画显示窗口再现了整个船舶上的灯光显示, 面板窗口用于从局部详细显示灯光状态, 报警器状态和模拟设备系统的运行情况. 表格显示窗口用于记录各个系统的运行信息, 以文字的形式表现, 便于信息之间的对比和分析. 最后, 还可以利用打印功能实现信息的保存, 为资料的存储作准备. 其效果图如图 3 所示.

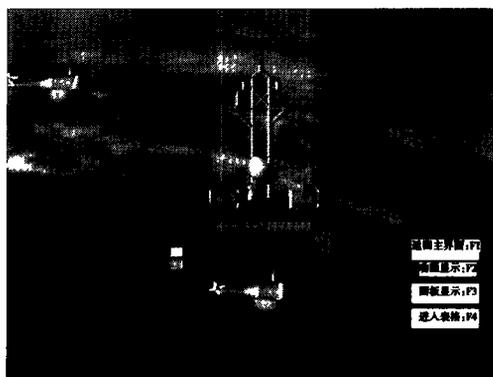


图 3 效果仿真图

3 结束语

该系统成功地利用 OpenGL 技术来模拟灯光显示, 对引导直升机的起飞、着陆、滑行具有一定的实用价值.

参 考 文 献:

- [1] 白建军, 朱亚平, 梁 会, 等. OpenGL 三维图形设计与制作[M]. 北京: 人民邮电出版社, 1999.
- [2] 向世明. OpenGL 编程与实例[M]. 北京: 电子工业出版社, 1999.
- [3] 彭晓明, 王 坚. OpenGL 深入编程与实例揭秘[M]. 北京: 人民邮电出版社, 1999.
- [4] BATES J, TOMPKIN T. 实用 Visual C++ 6.0 教程[M]. 北京: 清华大学出版社, 2000.
- [5] 李 颖, 薛海斌, 朱伯立, 等. OpenGL 技术应用实例精髓[M]. 北京: 国防工业出版社, 2001.

[责任编辑: 李雷莲]