

上海电力学院

虚拟现实技术 课程设计报告



题目 3D 人脸渲染

姓名 张皓霖

专业 电子信息工程

班级 2015071

学号 20151340

一、目的

- (1) 熟悉 Opengl 工具
- (2) 学会在 Ubuntu 中运行 OpenGL 示例
- (3) 了解 OpenGL 并用自己的图片渲染 3D 人脸模型

二、环境

VMware Workstation Pro—Ubuntu
Win10 VS2013

三、原理及主体

1、加载照片和纹理

```
void FaceTexture(void)
{
    image = auxDIBImageLoad("face2.bmp");

    if (!pdata != NULL)
        free(pdata);
    pdata = (BYTE*)malloc(256 * 256 * 3 * sizeof(BYTE));
    if (!pdata)
        exit(0);
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
    gluScaleImage(GL_RGB, image->sizeX, image->sizeY, GL_UNSIGNED_BYTE, image->data,
        256, 256, GL_UNSIGNED_BYTE, pdata);
    glTexImage2D(GL_TEXTURE_2D, 0, 3, 256, 256, 0, GL_RGB, GL_UNSIGNED_BYTE, pdata);
}
```

2、加入光影效果

```
void initlights(void)
{
    GLfloat ambient[] = { 0.2, 0.2, 0.2, 1.0 };
    GLfloat position[] = { 0.0, 0.0, 1.0, 1.0 };
    GLfloat mat_diffuse[] = { 0.8, 0.8, 0.8, 1.0 };
    GLfloat mat_specular[] = { 0.0, 0.0, 1.0, 1.0 };
    GLfloat mat_shininess[] = { 100.0 };

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);

    glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
    glLightfv(GL_LIGHT0, GL_POSITION, position);

    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
}
```

3、提高图片质量，加入抗锯齿效果

```
//OpenGL设定
glEnable(GL_BLEND); //启用混合功能，将图形颜色同周围颜色相混合
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

glEnable(GL_POLYGON_SMOOTH); //多边形抗锯齿
glHint(GL_POLYGON_SMOOTH, GL_NICEST);

glEnable(GL_LINE_SMOOTH); //线抗锯齿
glHint(GL_LINE_SMOOTH_HINT, GL_NICEST);

glEnable(GL_POINT_SMOOTH); //点抗锯齿
glHint(GL_POINT_SMOOTH, GL_NICEST);
```

4、glTexParameter: 用来设置纹理参数，通常需要设置的参数有 4 个：第一个是 GL_TEXTURE_MAG_FILTER，指当目标大于当前纹理图像时如何处理；第二个是

GL_TEXTURE_MIN_FILTER, 指当目标小于当前纹理图像时如何处理; 第三个是 GL_TEXTURE_WRAP_S, 指当纹理坐标的第一维坐标值大于 1.0 或者小于 0.0 时如何处理。纹理图像的坐标范围是为 (0,0) ~ (1,1), 左下角是 (0,0), 右上角是 (1,1)。
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE): 设置纹理跟材质的融合方式, 不同的设置可以产生不同的效果, 默认为 GL_REPLACE, 即只是用纹理, 覆盖模型的材质。

```
glTexImage2D(GL_TEXTURE_2D, 0, 3, 256, 256, 0, GL_RGB, GL_UNSIGNED_BYTE, pdata);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
```

5、保存纹理并载入, glTexImage2D: 用于载入一个二维的纹理。glTexCoord2f: 用于绘制图形时指定纹理的坐标。

```
void display(void)
{
    glTexCoord2d(0.401250, 0.728333); glVertex3f(-0.200000f, -0.461000f, -0.024000f);
    glTexCoord2d(0.565000, 0.707083); glVertex3f(0.123000f, -0.410000f, 0.063000f);
    glTexCoord2d(0.551875, 0.732083); glVertex3f(0.100000f, -0.461000f, 0.050000f);
    glTexCoord2d(0.596875, 0.728333); glVertex3f(-0.200000f, -0.461000f, -0.024000f);
    glTexCoord2d(0.627083, 0.727708); glVertex3f(0.246000f, -0.461000f, -0.000000f);
}
```

6、视角

```
void reshape(int w, int h)
{
    glViewport(0, 0, (GLsizei)w, (GLsizei)h);
    gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 1.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (w <= h)
        glOrtho(-1.5, 1.5, -1.5*(GLfloat)h / (GLfloat)w,
                1.5*(GLfloat)h / (GLfloat)w, -10.0, 10.0);
    else
        glOrtho(-1.5*(GLfloat)w / (GLfloat)h,
                1.5*(GLfloat)w / (GLfloat)h, -1.5, 1.5, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

7、交互

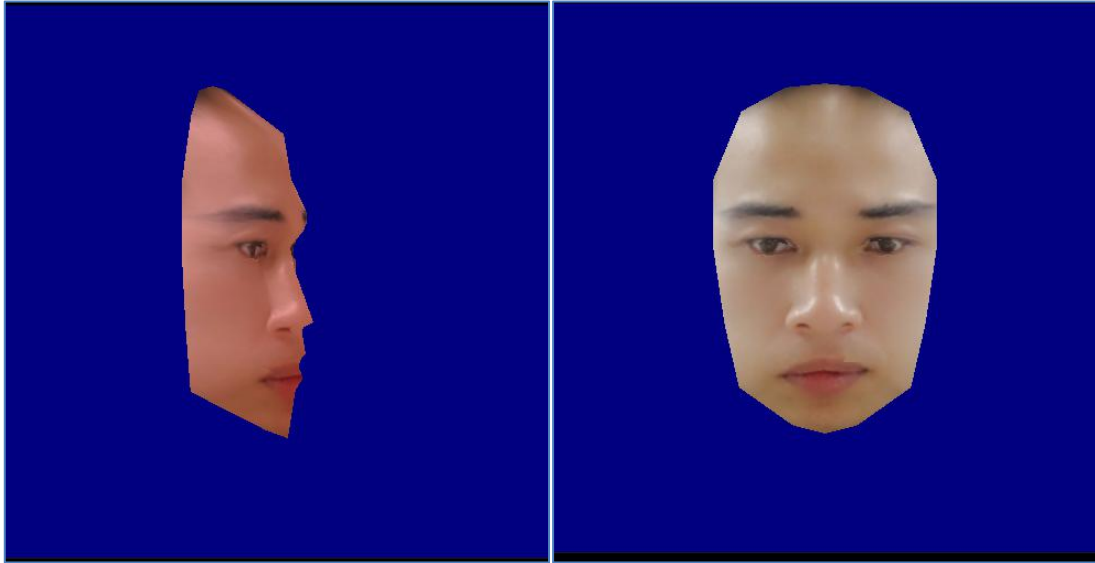
```
void mouse(int button, int state, int x, int y)
```

```
void keyboard(unsigned char key, int x, int y)
```

8、主函数调用各子函数模块

```
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow(argv[0]);
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(mouse);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}
```

四、效果



五、总结

本次课程设计，是我第一次在全老师的指引下使用 linux 系统（我后面也顺利入手并利用 linux 系统也在 github 上查找了资料完成了信息对抗与安全的课程设计），其次，是入门和了解 OpenGL 工具并使用 OpenGL 完成一些简单到复杂的图像设计和处理，使我的眼界和知识面有了更大的拓展。这学期最大的收获是，全老师给了很多方面的指导和指引，未来的学习中，我更多的会去熟练使用 linux 系统和一些学习工具，利用好 github 上面丰富的开源资源，在属于自己的领域上做更深入的研究和学习。