

3DS MAX 模型在 OPENGL 中的读取与重现

郭 景,雷 鸣

(天津工业大学 计算机与自动化学院,天津 300160)

摘要:介绍了 3DS MAX 文件的存储格式,在 VC++ 下的 OpenGL 编程中完成了 3DS MAX 模型的读取与重现。利用现有的三维图形软件(如 3DS MAX)建立物质模型,再用 OpenGL 对其进行控制,随心所欲地实现虚拟现实。

关键词:OpenGL 3DMAX 存储格式 法向量

中图分类号:TP311.134.3

文献标识码:B

OpenGL 即开放性图形库 (Open Graphic Library) 是一个三维的计算机图形和模型库。它独立于操作系统和硬件环境,适用于从个人计算机到工作站的广泛的计算机环境。OpenGL 在三维真实感图形制作中具有优秀的性能,用这个图形库不仅能方便地制造出有极高质量的静止彩色图像,还能创造出高质量的动画效果。借助 Windows 编程环境还可以对模型的人机交互进行控制。由于其开放性和高度的可重用性,目前已成为业界标准。

收稿日期:2002-03-27

但 OpenGL 并没有提供三维模型的高级命令,它也是通过基本的几何图元——点、线及多边形来建立三维立体模型的。目前有许多优秀的三维图形软件如 3DS MAX, AUTOCAD, 可以方便地建立物体模型,但又难以对其进行控制^[1],比如 3DS MAX 的动画是基于关键帧而不是人机交互的。把这些模型转换成 OpenGL 程序,再对其进行控制则是一种比较理想的方法。在建模阶段可以利用三维图形软件建立模型,避免在 OpenGL 中用点、线

数值所对应的隶属函数概念作为模糊评价结果。具体流程如图 6 所示。

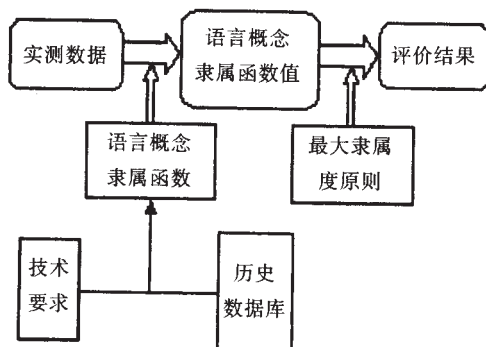


图 6 模糊评价流程图

要提高评价结果的正确性,必须对历史数据库(存放以往的测试数据及评价结果)及时更新;增加实测值个数,有利于进一步消除干扰数据,提高隶属函数值的准确度。

4 小结

为使变频节能供水设备的批量生产顺利进行,并保证产品质量,设计并实现了一套变频节能供水设备的计算机辅助测试系统。由于变频节能供水设备尚无国家标准,没有任何行业性通用标准,使得常规的测试评价手段无法利用。因此,结合该系统的实际情况,提出了以模糊测试模糊评价为核心的测试流程,从而保证了测试结果的可信度。该测试系统自投入使用以来,用户反映较好,实践证明该测试系统不论是测量精度还是可靠性都符合实际需要。

参考文献:

- [1] 宋阳. 可编程逻辑控制器及其在工程中的应用[M]. 北京:北京希望电脑公司,1996.
- [2] 朱麟章,蒙建波. 检测理论及其应用[M]. 北京:机械工业出版社,1997.
- [3] 杨伦标,高英仪. 模糊数学原理及应用[M]. 广州:华南理工大学出版社,1998.

和多边形来建立三维立体模型的繁重劳动；在控制阶段，则可以通过 OpenGL 程序实现对模型的人机交互式调整，实现了资源共享，也避免了重复劳动，从而可以随心所欲地利用现成的模型实现虚拟现实（Virtual Reality）。

1 3DS MAX 数据文件的存储格式

3DS MAX 默认的数据文件是 *.max 文件。*.max 结构复杂，数据量庞大，不仅包括模型网格结构的点和面的信息，还包括模型的颜色、光照、阴影、材质、贴图以及各种场景设置，并且文件以十六进制数存储，读取比较困难。除默认的 *.max 结构外，3DS MAX 还可以导出其他多种文件格式，包括：*.3ds（由 3ds/max 低版本生成的文件），*.ASC（ASC File Export 文件），*.ASE（ASCII Scene Export 文件），*.DFX（AutoCAD 文件），*.AI（Adobe Illustrator 文件）^[4]。以下介绍一种较简单直观的文件存储格式——*.ASC。

1.1 *.ASC 文件的存储格式

.ASC 文件以三角形网格形式存储 3DS MAX 模型，不仅满足 OpenGL 用多边形构造曲面时的顶点共面性，而且清楚明了。.ASC 文件格式如下：

```
Ambient light color: Red = 0.0000 Green = 0.0000 Blue = 0.0000
Named object: "Sphere01"

Tri - mesh, Vertices: 182    Faces: 360
Mapped
Vertex list:
Vertex 0:  X: -3.5601  Y: -26.0093  Z: 18.1208  U: 0.0000
           V: 1.0000
Vertex 1:  X: -3.5601  Y: -23.5822  Z: 33.4448  U: 0.0500
           V: 1.0000
...
Vertex 181: X: -3.5601  Y: 73.1694   Z: 18.1208  U: 0.6500
           V: 0.2000

Face list:
Face 0:  A: 0 B: 1 C: 2 AB: 1 BC: 1 CA: 1
Material: "1 - Default"
Smoothing: 1
Face 1:  A: 0 B: 2 C: 3 AB: 1 BC: 1 CA: 1
Material: "1 - Default"
Smoothing: 1
...
Face 359: A: 181 B: 161 C: 180 AB: 1 BC: 1 CA: 1
Material: "1 - Default"
Smoothing: 1
```

```
Named object: "Spot01"
Direct light
Position: X: 65.1007    Y: -0.0000    Z: 89.2617
Spotlight to: X: -48.9933    Y: 0.0000    Z: -8.7248
Hotspot size: 2463.7183 degrees
Falloff size: 2578.3101 degrees
Light color: Red = 1.0000    Green = 1.0000    Blue = 1.0000
```

其中，Ambient light color 是环境光设定值，以 RGB 三元组表示。从 Tri - mesh 段开始是模型网格数据，由点数、面数、点列表和面列表组成，是读取模型时的关键数据。而 3DS MAX 的场景信息，包括场景名称、渲染速度、场景背景和静态环境、材质信息、光照、阴影和颜色等参数，将忽略不读，这并不影响对模型的读取、重现和控制。因为所关心的模型形状的信息由 Vertex list 和 Face list 给出^[3]，其他场景信息在 OpenGL 中可以方便地重新设置和调整。

1.2 读取模型的关键数据

以 Sphere01 模型为例介绍读取模型的关键数据。Tri - mesh 是模型网格信息段开始的标志；Named object: "Sphere01" 是场景中一个物体的名称，同一场景中可以有多个物体；Vertices: 182 Faces: 360 表示模型有 182 个点及 360 个面；Face list 是点列表开始的标志，该段包括点号和点的坐标，例如 Vertex 0: X: -3.5601 Y: -26.0093 Z: 18.1208 表示点 0（-3.560 1，-26.009 3，18.120 8）；Face list 面列表包括面号和构成该面的顶点对应的点号，例如 Face 0: A: 0 B: 1 C: 2 表示三角形面 0 由点 0，点 1 和点 2 构成，顶点 A，B，C 分别对应点 0，点 1 和点 2，“AB”表示在点 A，B 之间存在一条自点 A 到点 B 的有向边，AB: 1 BC: 1 CA: 1 表示三角形 A→B→C→A。有了以上信息，就可以在 OpenGL 中读取该模型了。而贴图坐标 U 及 V，材质 Material 和光照参数 Named object，Direct light，Position，Spotlight to，Hotspot size，Falloff size，Light color 将在 OpenGL 中重新设定。

2 3DS MAX 模型的读取

在 OpenGL 中读取 3DS MAX 模型的步骤：①在 OpenGL 中建立自己的数据结构存储模型数据；②从 *.ASC 文件中读取数据将其存储到 OpenGL 程序中；③建立 OpenGL 绘制模型的显示列表。

2.1 在 OpenGL 中建立模型数据结构

场景中一个物体的数据存储结构：

```
typedef struct Object_Type{
```

```
    int Vertex_num;
```

```
    int Face_num;
```

```
    float Vertex[ ][ ];
```

```
    float Face[ ][ ];
```

```
    Normal[ ][ ];
```

```
}OBJECT;
```

其中, Vertex_num, Face_num 存储点数、面数;

Vertex[i][j] 存储点列表, 下标 i 对应点号, j 为 $0 \sim 2$, 对应坐标 (x, y, z) 的三个分量; float Face[i][j] 存储面列表, 下标 i 是面号, j 为 $0 \sim 2$, 存储三角形三个顶点对应的点号, Face[i][0] 对应存储三角形顶点 A , Face[i][1] 对应顶点 B , Face[i][2] 对应顶点 C ; Normal[][] 用于存储面的法向量。

如果场景中有多个物体, 可以用 OBJECT 类型的数组存储。

2.2 从 *.ASC 文件中读取数据

```
void CRead3dmaxView::GetObject(LPCTSTR filename)
```

```
{
    CStdioFile my_file;
    int vn, fn;
    int flag = 1; //文件是否结束
    vn = -1;
    fn = -1;
    Object_num = 0;
    CString line; //文件的一行
    while(flag == 1)
    {
        if(! my_file.ReadString(line))
            flag = 0; //读取文件失败或文件结束
        if (Find("Named object")) //读取文件一行, 判断是否为 Named
            object
                //Find ()为自定义函数

        Object_num = Object_num + 1;
        if(Find("Tri - mesh"))
        {
            Vertex_num = get_Vertex_num(); //自定义函数
            Face_num = get_Face_num(); //自定义函数
        }
        if (Find("Vertex list: "))
        {
            vn = vn + 1; //点数加 1
            if (Find("Vertex: "))
            {Vertex[vn][0] = get_coordinate_x(); //自定义函数
              Vertex[vn][1] = get_coordinate_y(); //自定义函数
              Vertex[vn][2] = get_coordinate_z(); //自定义函数
```

```
            }
        }
    }
    if (Find("Face list"))
    {
        fn = fn + 1; //面数加 1
        if (Find( "Face ")
        {
            Face[fn][0] = get_point_A(); //自定义函数
            Face[fn][1] = get_point_B(); //自定义函数
            Face[fn][2] = get_point_C(); //自定义函数
        }
    }
} //end while
my_file.Close();
}
```

2.3 建立 OpenGL 显示列表

```
void CRead3dmaxView::Makelist()
```

```
{
    int i, j;
    glNewList(point, GL_COMPILE); //建立点列表
    glBegin(GL_POINTS); //调用 OpenGL 函数画点
        for(i = 0; i <Vertex_num; i + + )
        {
            glNormal3f(Vertex[i][0], Vertex[i][1], Vertex[i][2]);
                //点的法向量
            glVertex3f(Vertex[i][0], Vertex[i][1], Vertex[i][2]);
                //点的坐标
        }
    glEnd();
    glEndList(); //点列表结束
    glNewList(box, GL_COMPILE); //建立面列表
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    glBegin(GL_TRIANGLES); //调用 OpenGL 函数画三角形面
        for(i = 0; i <Face_num; i + + )
        {
            glNormal3f(Normal[i][0], Normal[i][1], Normal[i][2]);
                //面的法向量
            for(j = 0; j <3; j + + )
                glVertex3f (Vertex [Face[i][j]][0], Vertex [Face[i][j]][1], Vertex
                    [Face[i][j]][2]); //三角形的三个顶点的坐标
        }
    glEnd();
    glEndList(); //面列表结束
}
```

至此, 完成了从 *.ASC 文件中导入模型数据。

3 3DS MAX 模型的重现

在 OpenGL 中重现 3DS MAX 模型是通过在 OnDraw(CDC * pDC) 函数中调用显示列表实现

的。在重绘场景时除了进行视点——模型变换、投影变换和视口变换,进行必要的光照材质等设置外,还必须计算法向量。法向量在用多边形构造三维造型时是至关重要的,它与实体的光照模型密切相关。光线从不同入射角照射到模型表面时,显示结果有差别,如果忽略了法向量的计算,将不能重现模型的三维效果。

3.1 法向量的计算

对于一个曲面而言,法向量是与顶点密切相连的。曲面上某一点的法向量与顶点的一个邻域垂直。当邻域尺寸收缩为足够小时,原本属于曲面的邻域可以足够精确地被看成平面,从而有近似意义下的法线^[1]。

连续曲面 S 用显式表达: $S(p, q) = S[x(p, q), y(p, q), z(p, q)]$ 。

$$\text{写成含参数的形式: } \begin{cases} x = x(p, q) \\ y = y(p, q) \\ z = z(p, q) \end{cases}$$

式中: p, q 为曲面参数; x, y, z 均为可微参数。曲面法向量可用 $\frac{\partial S}{\partial p} \times \frac{\partial S}{\partial q}$ 表示。由于 OpenGL 的三维模型用多边形(三角形)近似拟合得到,所以法向量同样基于多边形(三角形)。例如:图 1 中 a 点的邻域是由四个三角形拼合而成,由于这些三角形不共面,故可将归一化后的法向量 n_1, n_2, n_3, n_4 作加权平均,得到点 a 的法向量。

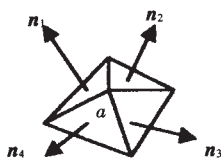


图 1

自定义函数 `void CRead3dmaxView::GetNormal(float * nx, float * ny, float * nz, int facename)`, 计算面号为 `facename` 的面的法向量。计算完成后,将法向量加入 OpenGL 的显示列表。在 OpenGL 中,用函数 `Void glNormal3f(bisdf) (Glbyte nx, Glbyte ny, Glbyte nz)` 指定法向量,用函数 `glEnable(GL_NORMALIZE)` 归一化法向量。经过归一化的法向量其模不受坐标平移、转动的影响。

3.2 在 OpenGL 中重绘模型

读取了模型的网格数据,计算了法向量之后,通

过以下步骤就可以实现模型的重绘:

① `#include <gl\gl. h>`

`#include <gl\glu. h>`

`#include <gl\glaux. h>`

② 在 `Read3dmaxView::OnCreate(LPCREATES TRUCT lpCreateStruct)` 函数中设置像素格式。

③ 在 `CRead3dmaxView::OnSize(UINT nType, int cx, int cy)` 函数中进行投影变换和视口变换。

④ 自定义函数 `CRead3dmaxView::Drawscene()`, 进行灯光、材质、颜色等设定。

⑤ 在 `CRead3dmaxView::OnDraw(CDC * pDC)` 函数中实现模型重绘:

```
{
    wglMakeCurrent(pDC -> m_hDC, hglrc);
    Drawscene();
    GetObject();
    Makelist(sphere);
    glCallList(shpere);
    SwapBuffers(pDC -> m_hDC);
    wglMakeCurrent(pDC -> m_hDC, NULL)
}
```

运行程序,就可以在 OpenGL 中得到用 3DS MAX 建立的模型。

4 结束语

使用该方法可以有效地利用现有的三维建模软件,成功地实现 3DS MAX 模型在 OpenGL 中的应用。通过 OpenGL 拾取反馈模式,还能方便地对模型进行交互式控制。这样,就可以在 OpenGL 中随心所欲地使用优秀的 3DS MAX 模型。本方法有多方面的应用,如纺织面料三维着装系统的设计、室内装饰系统的设计、服饰配件的效果设计等,它将避免大量的重复劳动,有效地提高工作效率。

参考文献:

- [1] 白建军,朱亚平,梁辉,等. OpenGL 三维图形设计与制作[M]. 北京:人民邮电出版社,1999.
- [2] 尚游,陈研涛. OpenGL 图形程序设计指南[M]. 北京:水利水电出版社,2001.
- [3] 读取 DFX 格式文件[J/OL]. 软件世界,2000(5).
- [4] 张颖,高征. 3DS MAX 基础教程[M]. 北京:北京希望电子出版社,2001.