

基于 OpenGL 的 PNG 纹理映射的实现^①

Realization of PNG Texture Mapping Based on OpenGL

尹航 李义杰 孙健超 (辽宁工程技术大学 电子与信息工程学院 辽宁 葫芦岛 125105)

摘要: 科学计算可视化, 计算机动画和虚拟现实是现在计算机图形学的三个热点。而这三个热点的核心都是三维真实感图形的绘制。由于 OpenGL 具有跨平台性、简便、高效、功能完善等优点, 目前已经成为了三维图形制作方法的工业标准。因此对 OpenGL 的研究与应用是一个极其有意义的课题。本文通过对 OpenGL 的学习, 利用第三方库 CxImage 解决了 OpenGL 不能对 PNG 格式的图片进行纹理映射的难题。

关键词: 计算机图形学 开放式图形库 图象操作类库 可移植网络图形 纹理映射

1 引言

OpenGL 的英文全称是 "Open Graphics Library" 即 "开放的图形程序接口", 它是近几年发展起来的一个性能卓越的三维图形标准。OpenGL 提供了几百个图形函数, 开发人员可以利用这些函数来构造景物模型, 开发三维图形实时交互软件。OpenGL 具有强大的图形功能和良好的跨平台移植能力, 目前已被广泛应用于可视化技术、实体造型、CAD/CAM、模拟仿真等诸多领域^[1]。

纹理映射技术是 OpenGL 的一个重要的技术, 在进行模型转换和投影转换都能执行操作, 可用到所有的图元-点、线、多边形、位图和图像上。它的纹理映射函数功能较弱, 只能对 BMP 格式的图片进行纹理映射。本文主要讨论了在 VS2005 环境下利用 OpenGL 和第三方库 CxImage 对 PNG 格式的图片进行纹理映射的方法。

2 基本概念

2.1 纹理

纹理的定义有两种: 连续法和离散法。连续法把纹理定义为一个二元函数, 函数的定义域就是纹理空间。而离散法则是把纹理定义在一个二维数组中, 该数组表示纹理空间中行间隔和列间隔固定的一组网格点上的纹理值。网格点之间的其它点的纹理值可以通

过对相邻网格点上纹理值进行插值来获得。通过纹理空间与物体空间之间的坐标变换, 可以把纹理映射到物体表面。一般来说, 离散法是较为常用的纹理定义方式^[2]。

纹理的分类: 根据纹理的表现形式可以分为颜色纹理、几何纹理和过程纹理。

(1) 颜色纹理。在物体表面绘制图案, 就是改变物体表面相关部分的反射和透射系数, 采用这种方法得到的纹理称为颜色纹理。颜色纹理的实质是将一个二维纹理模式映射到物体表面上, 这二维纹理模式是定义在一个平面区域上, 平面区域的每一个点, 定义一个灰度或颜色值, 该平面区域称为纹理空间, 该方法称为纹理映射。

(2) 几何纹理。颜色纹理只能在光滑物体表面上绘制事先定义的花纹图案。几何纹理用于表现细微的凹凸不平的景物表面, 此细微的表面凹凸数据结构的表示非常困难, 可用一种特殊的算法模拟, 显示具有纹理的景物图像。

(3) 过程纹理。用一些简单的可解析的数学模型来描述复杂的自然纹理细节, 即用过程方式将纹理空间的值映射到景物的表面, 生成的三维纹理称为过程纹理。

2.2 PNG 和 CxImage 库

PNG 的全称 Portable Network Graphics(可移

^① 收稿时间: 2008-10-30

植网络图形), Fireworks 的默认存盘格式就是 PNG。这种新的图片文件格式集合了 GIF 和 JPG 格式的优点, 从目前来看, 它必将在未来的 Web 网页中大显身手, 成为 Web 中的主要图片格式。PNG 支持背景透明、图形渐进和动画, 而且是种无损压缩格式, 压缩比例要大大超过 LZW 等传统的图片无损压缩算法。同时它也支持几百万种颜色, 既适合艺术线条, 也适合照片。

下面用表格的方式列出三种常用图片格式的主要特性:

表 1 三种格式图片的特性对比

特性	GIF	JPG	PNG
背景透明	是	否	是
图形渐进	是	否	是
支持动画	是	否	是
无损压缩	是	否	是
上百万种颜色	否	是	是
适用于线条	是	否	是
适用于照片	否	是	是

从这个表格也能看出 PNG 格式的优势, 随着网络带宽的提升, PNG 必将成为网络图片的主流格式, 而且最新 IE7.0 的一项更新就是支持透明 PNG, 使浏览器可以显示覆盖图。

由于 OpenGL 读取图形文件做纹理的函数的功能很弱, 只支持 BMP 图片, 如果要读取其它格式的纹理, 就需要用到第三方函数库 CxImage 了。CxImage 类库是个优秀的图像操作类库。它能够快捷地存取、显示、转换各种图像。CxImage 类库是完全免费的, 而且 CxImage 类库的源代码是公开的。相对于那些封装好的图像库和 GDI+来说, 这一点使读者能够进一步学习各种编解码技术, 而不再浮于各种技术的表面^[3]。

3 PNG纹理映射的实现

3.1 纹理映射的原理

OpenGL 纹理映射的原理与 DirectX 纹理映射的原理相同, 都是在定义完纹理对象之后, 通过先描画几何形状, 然后将图片贴到几何形状的表面, 本实验也采用了这个原理^[4]。

3.2 纹理映射的执行步骤

在本实验中完成 PNG 纹理映射的需执行的步骤:

(1)加载资源。资源指的是所要显示的图片, 在这里是 PNG 格式的图片, 可以将每一个图片定义一个 ID 号, 然后通过 FindResource 和 LoadResource 将其

加载进来, 具体代码如下:

```
newImage[0].LoadResource(FindResource(NULL,MAKEINTRESOURCE(IDR_PNG),_T("PNG")),CXIMAGE_FORMAT_PNG);
```

这里的 newImage[0]就是 CxImage 类的一个具体实例化的对象数组中的一员, 用来存储 PNG 图片资源, 以生成可用的纹理。

(2)定义纹理。纹理通常被认为是二维的, 但纹理也可以是一维或三维的。通过函数 glTexImage2D() 指定一个二维纹理, 其中包含了纹理图像的大小、纹理图像数据的数据格式和数据类型以及存储在内存中的图像数据指针等。其函数原型为:

```
void glTexImage2D( GLenum target, GLint level, GLint components, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid *pixels );
```

(3)控制纹理。用来说明纹理以何种方式映射到三维模型的表面上。一种方法是, 将纹理颜色作为最终的颜色, 另一种方法是, 使用纹理来调整片元的颜色值, 还有一种就是将一种常量颜色和片元混合起来。OpenGL 中提供了相关的函数为 glTexParameter()。下面的两段代码告诉 OpenGL 在显示图像时, 当它比放大得原始的纹理大或缩小得比原始得纹理小时 OpenGL 采用的滤波方式。通常这两种情况下都采用 GL_LINEAR。这使得纹理从很远处到离屏幕很近时都平滑显示。使用 GL_LINEAR 需要 CPU 和显卡做更多的运算。

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
```

(4)描画矩形。由于 PNG 纹理映射的原理是先画一个矩形, 然后将所要生成纹理的 PNG 图片贴到矩形上, 所以要想完成纹理映射, 必须先描画一个矩形。

为了将纹理正确的映射到四边形上, 必须将纹理的右上角映射到四边形的右上角, 纹理的左上角映射到四边形的左上角, 纹理的右下角映射到四边形的右下角, 纹理的左下角映射到四边形的左下角。如果映射错误的话, 图像显示时可能上下颠倒, 侧向一边或者什么都不是。

glTexCoord2f 的第一个参数是 X 坐标。 0.0f

是纹理的左侧。0.5f 是纹理的中点，1.0f 是纹理的右侧。glTexCoord2f 的第二个参数是 Y 坐标。0.0f 是纹理的底部。0.5f 是纹理的中点，1.0f 是纹理的顶部。

所以纹理的左上坐标是 X: 0.0f, Y: 1.0f，四边形的左上顶点是 X: -1.0f, Y: 1.0f。其余三点依此类推。

具体代码如下:

```
glBegin(GL_QUADS);
glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0f, -1.0f, 1.0f);
glTexCoord2f(1.0f, 0.0f); glVertex3f( 1.0f, -1.0f, 1.0f);
glTexCoord2f(1.0f, 1.0f); glVertex3f( 1.0f, 1.0f, 1.0f);
glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0f, 1.0f, 1.0f);
glEnd();
```

(5)启用纹理映射。绘制场景之前需要启用纹理映射。要启用和禁用纹理映射,可以调用函数 glEnable() 和 glDisable()。调用这些函数时可以使用一维、二维、三维和立方图纹理映射。由于 PNG 图片是具有透明效果的,要想显示其透明的效果,就要用到 OpenGL 中混合这个技术和一些有关 Alpha 的设定的函数,代码如下:

```
glEnable( GL_BLEND );
glBlendFunc( GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA );
glEnable(GL_TEXTURE_2D);
glDisable(GL_TEXTURE_2D);
glDisable(GL_BLEND);
```

(6)纹理坐标和绘制场景。在绘制带纹理的场景时,必须同时提供物体坐标和纹理坐标。物体坐标决定在屏幕上的什么地方对特定顶点进行渲染,纹理坐标控制纹理图像中的纹理怎样映射到物体。纹理坐标可以是 1、2、3、4 维的,通常也用齐次坐标来表示即(t,r,q)。在粘贴纹理之前,必须指定纹理和片元之间的对应关系。也就是说,在场景中绘制物体时,必须指定纹理坐标和几何坐标。调用的函数是 glTexCoord()^[5]。

3.3 纹理对象的管理和使用

纹理映射是把二维的纹理图象映射到三维物体表

面,其关键点就是建立物体空间坐标(x,y,z)与纹理空间坐标(s,t)之间的对应关系。为生成具有真实感的图形,利用纹理映射技术将复杂物体的图象粘贴到简单几何体的表面,置于场景中,在实时显示场景时,还可利用 3D 图形的平移、旋转能力,实现复杂物体随观察方向的改变而转动的效果。然而纹理加载的过程可能会影响程序运行速度,当纹理图像非常大时,这种情况尤为明显。如何妥善的管理纹理,减少不必要的开销,是系统优化时必须考虑的一个问题。还好,OpenGL 提供了纹理对象管理技术来解决上述问题。与显示列表一样,纹理对象通过一个单独的数字来标识。这允许 OpenGL 硬件能够在内存中保存多个纹理,而不是每次使用的时候再加载它们,从而减少了运算量,提高了速度。以本实验为例,使用纹理对象的步骤如下:

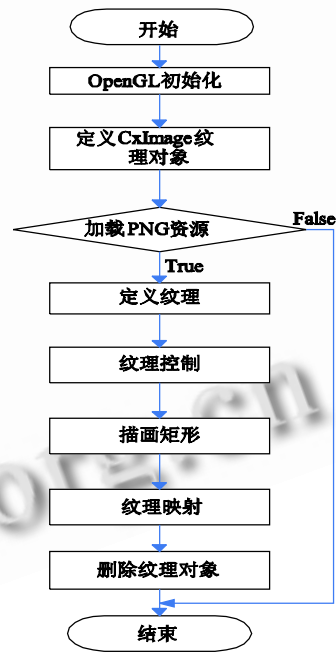


图 1 PNG 纹理映射的流程图

第一步:定义纹理对象

```
Coast int TexNumber 20;
```

```
GLuint Texture[TexNumber]; //定义纹理对象数组
```

第二步:生成纹理对象数组

```
glGenTextures(TexNumber,&Texture[m]);
```

第三步:通过使用 glBindTexture 选择纹理对象,来完成该纹理对象的定义。

```
glBindTexture(GL_TEXTURE_2D,Texture[m]);
```

```
gluBuild2DMipmaps(GL_TEXTURE_2D, 4,
sizeX, sizeY, GL_RGBA, GL_UNSIGNED_BYTE,
plmage_RGBA);
```

第四步:在绘制景物之前通过 `glBindTexture`, 为该景物加载相应的纹理。

```
glBindTexture(GL_TEXTURE_2D,
Texture[m]);
```

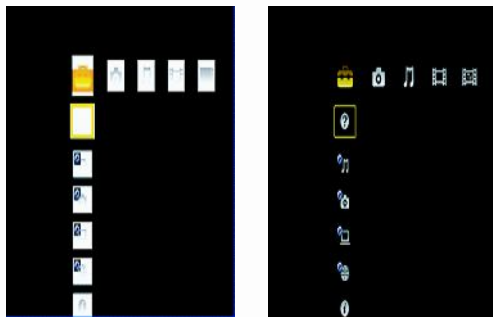
第五步:在程序结束之前调用 `glDeleteTextures` 删除纹理对象。

```
glDeleteTextures(TexNumber,Texture[m]);
```

这样就完成了全部纹理对象的管理和使用。结合纹理映射的执行步骤和纹理对象的管理与使用, 本实验执行的具体流程图如下:

4 实验结果对比分析

本实验分别用对 OpenGL 的 BMP 纹理映射和 PNG 的纹理映射进行了对比, 也对是否使用第三方库 CxImage 的 PNG 纹理映射前后进行了对比, 两者的对比如图 2 所示:



(a) BMP 纹理映射 (b) PNG 纹理映射

图 2 实验仿真前后的对比结果

本实验也对计算机性能情况进行了比较, 由于 PNG 是压缩图片, 在系统进程数和系统资源利用相同的情况下, 在显示相同内容的情况下, PNG 要比 BMP 等图片的大小要小的多, 在显示多纹理的时候, PNG 纹理映射的 CPU 和内存的利用率会明显的比 BMP 等纹理映射的 CPU 和内存的利用率低很多。对比情况如图 3 所示:

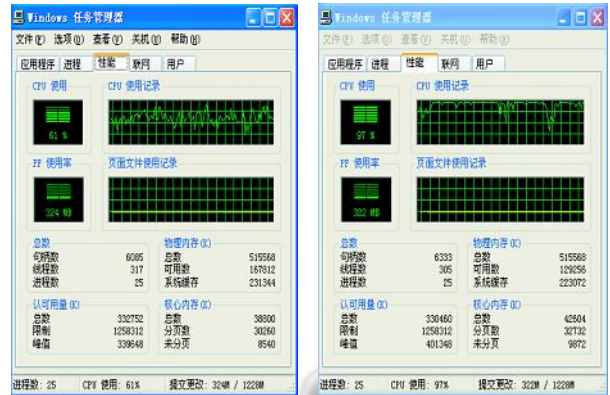


图 3 PNG 和 BMP 纹理映射系统性能对比图

5 结束语

通过对 OpenGL 的纹理映射技术的学习, 本文证明了利用第三方库 CxImage 在 OpenGL 中用 PNG 做纹理映射的方法是可行的, 不但能实现 PNG 图片透明等效果, 而且消耗系统资源也比较低^[6]。

参考文献

- 郭兆荣,李菁,王彦.Visual C++ OpenGL 应用程序开发.北京:人民邮电出版社,2006.
- Shreiner D, Woo M, Neider J, Davis T. OpenGL 编程指南(第四版).邓郑祥译.北京:人民邮电出版社,2005.
- 万斌.Visual C++ OpenGL DirectX 三维动画编程宝典.北京:希望电子出版社,2003.
- 和平鸽工作室.OpenGL 高级编程与可视化系统开发高级编程篇(第二版).北京:中国水利水电出版社,2006.
- 肖源源,王子牛.基于 VC++ 的 OpenGL 纹理映射技术的研究与实现.贵州大学学报(自然科学版), 2008,2: 158-160.
- 汤彬.基于 OpenGL 的纹理映射研究.上海:实验室研究与探索,2006,5:576-579.
- 张洁,刘小琴.基于 OpenGL 的纹理映射的实现.运城学院学报,2007,25(5):55-56.