

文章编号: 1000-582x(2000)05-0062-05

使用 OpenGL 开发实时三维应用程序

徐受容

(西南航天职工大学, 重庆 400021)

TP391.41

摘要: 计算机三维图形技术已发展到了实时阶段, 怎样有效地开发实时三维应用系统, 是目前较为引人注目的话题。笔者先后开发了数个 OpenGL 三维应用软件, 从中总结出了实现实时三维应用系统的不少关键技术, 包括实用的初始手段、分类分批建模、纹理、实时用户浏览控制及其位置处理等。这些技术已应用于实时电视特效播出系统、道路可视化设计/评价系统等。笔者将开发中总结的典型作法与经验作一个方法性概述。

关键词: 三维引擎; 纹理; 位图; 显示列表

中图分类号: TP 317.4

文献标识码: A

OpenGL

计算机三维图形技术

随着计算机技术尤其是计算机图形、图像处理技术的飞速发展, 在计算机屏幕上实时观察建筑、机械设计、城市的三维效果已不是梦想。笔者使用 OpenGL 设计实现了不少三维图形程序, 积累了很多经验。在本文中, 笔者将探讨利用 OpenGL 设计实时三维图形软件的方法。

1 模型

当前用于表示形体的模型主要有线框、曲面和实体 3 种。

线框模型是 CAD/CAM 领域中最早用来表示形体的模型, 至今仍在广泛应用。它结构简单、易于理解, 是曲面和实体模型的基础, 但不能用于处理计算机图形学和 CAD/CAM 中的多数问题, 如剖切图、消隐图、明暗色彩图、物性分析、加工处理等。

曲面模型用有向棱边围成的部分定义形体表面, 由曲面集合定义形体。曲面模型在线框模型的基础上, 增加了有关面边(环边)信息以及表面特征、棱边的连接方向等内容, 从而可以满足面面求交、线面消隐、明暗色彩图、数控加工等应用问题的需要。但在此模型中, 形体究竟存在于表面的哪一侧则没给出明确的定义, 因而在物性计算、有限元分析等应用中, 该模型在形体的表示上仍然缺乏完整性^[1]。

实体模型则明确定义了表面哪一侧存在着实体。在许多的 3D 引擎中, 一般都采用可让程序员根据法线矢量的方向决定形体的可视外侧, 或者依建模时的有向边的方向以右(左)手法则决定表面外侧。OpenGL 甚至支持双面着色 - 在形体表面的拓扑关系存在不相一致的情形时, 不论内外面均予以着色, 这对于很复杂的建模情况下无法或很难保证拓扑关系一致性时就很有作用^[4,5]。

2 3D 引擎——OpenGL

这里所讨论的是开发专业级的实时三维应用, 主要的着眼点就是 OpenGL。OpenGL 的前身是 SGI 公司为其图形工作站开发的 IRIS GL (IRIS Graphics Library), 是一个工业标准的三维计算机图形软件接口。其功能可划分为以下几类^[5]:

- 基本图素(如点、直线、多边形、三角形、三角形网络、矩形、圆和圆弧、字符、曲线和曲面以及读写像素。)
- 坐标变换(旋转、平移、比例变换以及窗口视图变换和裁剪, 同时支持用户定义的各种变换。)
- 设置属性和显示方式(定义选择线型、填充图案、字体和光标, 可设置 RGB 和颜色表两种选色方式以及明暗效果、双缓冲、各种位图等多种绘图方式。)

· 收稿日期: 2000-04-27

作者简介: 徐受容(1964-), 女, 四川成都市人, 西南航天职工大学副教授, 硕士。主要从事计算机软件方面的科研和教学工作。

· 输入输出处理。

· 真实图形显示(消除隐藏线、面、光照处理和深度排队。)

GL 运行于 UNIX 操作系统,与其它三维图形标准相比有以下特点:

· 图元丰富 除具有一般图元外,还有 B 样条曲线、Bezier 曲面、NURBS 曲面等。

· 颜色 GL 具有 RGB 和颜色索引表两种方式,有 Gourand 和 Phong 光照模型,使实体表面的亮度与色彩变化柔和。

· Z 缓冲技术 在每个像素上附加一个 24 位或 48 位的表示 Z 值的缓冲存储,这对曲线曲面的消隐、亮度随深度变化的处理,提高图形处理效率等具有重要作用。

· 光源 光源的强度、颜色、物体的反射方向、镜面反射系数、漫反射系数等都影响到一定光源照射下物体的最终显示效果。GL 提供了充分的光源处理能力,使用户可以得到非常生动的图像。

为了改进 IRIS GL 的可移植性,SGI 与 Microsoft 共同开发了 OpenGL。OpenGL 是输出到图形硬件的一个三维图形应用程序编程接口,可工作于网络模式之下。在多台机器互连的网络环境中,OpenGL 程序能以客户/服务器模式运行,因此即使在不同类型的计算机上运行客户端或服务端程序,OpenGL 程序也能正确运行。如果 OpenGL 程序不在客户/服务器方式下工作,则该机器既是客户也是服务器。OpenGL 被设计成独立于硬件和操作系统平台,流水线工作方式可以在不同的硬件平台上实现。用户可以方便地利用这个图形库,创建出接近光线追踪的高质量静止或动画的三维彩色图像,而且速度要比光线追踪算法快一个数量级。在不同的平台上的实现就结合了该平台的应用程序接口的特点,但是其内核级接口是不变的。另外,OpenGL 没有提供高级命令函数来定义复杂的 3D 物体,如汽车、飞机的部件等,程序员必须使用其建模命令以几何图元(Geometric Primitives)的点、线和多边形来构造所要得到的模型,虽然其高级扩展库提供了一些复杂物体的定义功能。该做法的目的主要出于提高 OpenGL 的运行效率和性能的考虑。

1992 年 7 月,SGI 发布了 OpenGL 的 1.0 版本,后来与 Microsoft 合作共同开发了 Windows NT 下的新版本。1995 年 12 月,由 OpenGL ARB(Architecture Review Board) 批准了 OpenGL 1.1 版本,这一版本的 OpenGL 性能得到了加强并引入了一些新功能,包括:在增强元文件中包含 OpenGL 调用,改进打印机支持,

顶点数组的新特征,提高顶点位置、法线、颜色及色彩指数、纹理坐标、多边形边缘标识的传输速度,引入新的纹理特征。

1999 年 SGI 推出的可视化 NT 工作站以数倍于普通 PC 的总线和图形子系统带宽大大超越了所有同类 PC 级图形工作站的性能,其主力应用的基础就是 OpenGL 1.2——一个更加强大的 3D 内核。

OpenGL 在专业应用上具有很大的优势,当前出色的三维软件大都支持 OpenGL,如 3DS MAX、Maya、Lightscape 等,它们需要专业级的图形能力、高稳定的可靠性,这是目前 PC 平台上其它图形引擎难以做到的。

作为专业级的图形引擎,OpenGL 的着色质量和系统稳定性强于微软的 DirectX。

使用 OpenGL 设计三维图形应用软件,除了一个应用软件所特有的与用户交互、数据计算等部分之外,与 OpenGL 相关的主要步骤有:

1) 构造几何要素(点、线、多边形、图像、位图),均由顶点(Vertexes)来定义,创建对象的数学描述;2) 在三维空间中放置对象,选择有利的场景观察点;3) 计算对象的颜色,这些颜色可能直接定义,或由光照条件及纹理间接给出。4) 光栅化,把对象的数学描述和颜色信息转换到屏幕上的像素,同时还可能执行消隐及其它对像素的操作。

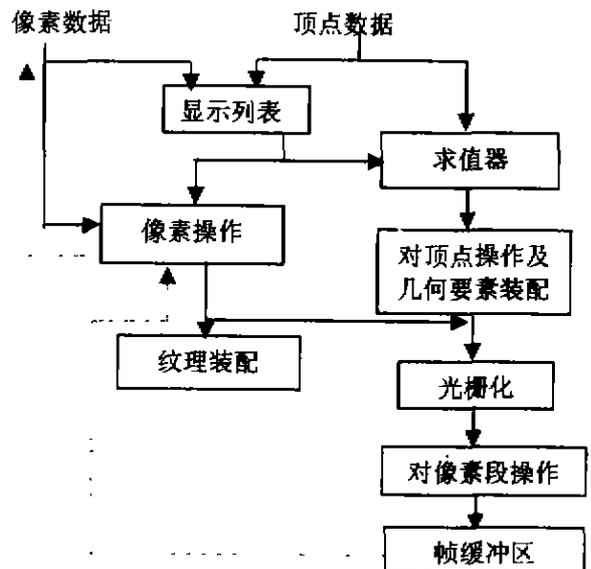


图 1 OpenGL 的处理流程

3 初始化及建模

在实时三维程序中对建模数据作一定处理之后,就应该作 OpenGL 的初始化及使用处理后的建模数据

创建三维模型。整个典型的处理流程如下:

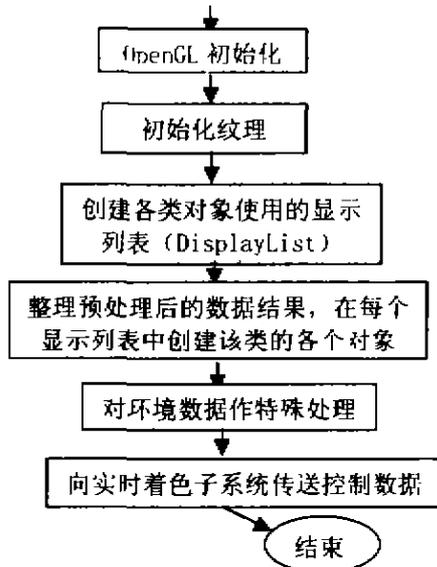


图2 初始化处理流程

3.1 初始化

一般地,对 OpenGL 的初始化工作主要有:

1) 置着色设备表面的像素格式

在 Windows 下,典型的初始化设置可有:

- 支持双缓冲——OpenGL 采用硬件或软件来提供两个完整的颜色缓冲区,当一个缓冲区显示时另一个用于绘制下一个场景。绘制完成后,两个缓冲区互相交换。且只有每一帧绘制完后才可显示出来。只要这个工作足够快,屏幕上就会出现连续的画面。对于需要实时着色的系统,一般都要设置双缓冲。

- 支持 Alpha 通道的 24 位色深的 32 位图像格式^[2,3]——指数模式显然已不适合动态且真实度要求很高的场合。另外,具有 Alpha 通道的 32 位像素格式由于具有 4 字节的可对齐性在处理上比 24 位像素格式(3 字节)更方便、快速,而且可用于后继的 Alpha 混合、反走样、雾化等处理。

- 允许深度缓冲(Z 缓冲)——由图形硬件或软件计算绘制的每个面对应于窗口中没有被遮挡的表面的一组像素值,另外,从观察者眼睛到物体表面的距离也同时被计算出来。当使用深度缓冲技术时,在像素被画出之前,首先比较存储在深度缓冲区中的深度值,如果新的像素更接近于视点,则新的颜色和深度值就取代缓冲区中的那些值,反之则被丢弃。这种消隐技术以占用 Z 缓冲器为代价提高系统运行性能。

- 设置合适的场景裁剪范围等。

2) 初始化纹理位图

将贴图用位图以设备无关位图(DIB)格式作尺寸归整(OpenGL 要求贴图位图的宽高必须为 2 的幂,在

开发中笔者实现了处于中间服务层的一个处理图像的对象库 GK,使用 GK 中实现的一个较精确、快速的图像插值再取样算法 ZoomBits 重新计算调整尺寸后的图像),然后设置合适的贴图参数以便 OpenGL 在场景渲染时使用。GK 在笔者开发的多个系统中都被广泛使用。

纹理是包含颜色、Alpha 值、亮度数据的数字图像的矩形数组,这一数组中的元素称为纹素。纹理映射可以把纹理图像粘贴到曲面上,并使纹理在变换时与所附着的曲面保持适当关系。纹理映射可用于所有几何要素,但仅能在 RGBA 模式下使用。

可以把纹理颜色作为像素的最终颜色,即使用纹理覆盖原有的像素段。也可以把光照与纹理化的效果结合起来,调整或放大像素段的颜色。另一个使用方法是基于纹理值的一个固定颜色与像素段颜色混合(Blend),不论采用何种混合方法都会带来附加的大量运算。因此,对于自己要实现的系统,应该在着色质量与着色速度上做出取舍后,决定采用何种纹理处理方式。

根据笔者的经验,在场景复杂且实时程度要求较高时,采用如下的方法:仅使用各一份纹理素材而不使用多重尺寸纹理以减轻 OpenGL 映射纹理时的存储开销,采用简单的纹理坐标映射方式减轻纹理映射的计算量和印花模式避免混合操作。实践证明,着色出的画面质量基本可以接受。随着复杂程度的降低,机器性能的提升,可以逐步增加纹理的质量,以达到一个新的平衡。

3.2 预编译显示列表

显示列表是预先存储并经 OpenGL 动态编译过的 OpenGL 命令,在重绘每一场景时调用显示列表能提高执行效率,尤其是在网络上,从而提高整个应用系统的性能。将需要大量计算的某些操作放入显示列表中对性能的提高尤为明显。当然,对于太小的显示列表,由于系统开销的原因可能体会不到其性能上的提升。OpenGL 建议用于优化的最大可能性有:矩阵操作、光栅位图和图像、光线及材质属性以及光照模式、纹理、多边形样条模式。

根据笔者的设计经验,最好将三维模型及其数据分类组织,创建各对象类的显示列表,每个显示列表都按照其对象类的特点作不同的优化,以提高实时着色的性能。如对于需要纹理贴图的列表,在生成其每一个实体的建模命令时,均要计算其纹理映射坐标,而对于如柏油路面等不需纹理的对象列表,则仅处理其光感、材料属性参数即可。由于使用了显示列表,就不必

每次绘制场景时都做复杂的材质定义、纹理坐标计算等耗费机时的工作了,场景的绘制效率得以提高。

另外,如道路设计等 CAD 应用使用的大地坐标系为左手系,而在三维技术中大量使用的是右手系的世界坐标,因此,在建模时将预处理后的数据略作修正,即交换 X 与 Y 的坐标值,得到右手系坐标值。

3.3 其它优化

创建对象的 3D 模型有很强的经验性,影响最后画面质量及绘制速度的因素比较多。应根据所设计系统的具体情况和特点采取措施,典型的有:

- 保证四边形的方向恒定。即保证当物体从外面看时,所有表面上的四边形在相同方向上(即所有的顶点序为顺时针或都是逆时针序),保证这种正确性可以避免混合序的法线矢量乱向问题。同时,采用这种方法可以使 OpenGL 只着色物体外表面,以提高着色速度,经测试,速度可提高约 10%~20%。

- 为了避免四边形 4 个顶点可能不共面的情况,在向某些显示列表中追加建模命令时,内部可将其分解为 2 个三角形的建模命令。在 OpenGL 中,创建三角形或四边形比创建多边形效率会高一些。这样,在建模的质量与渲染着色的速度上得到一个比较好的折中。

- 一般来说建模数据中对主体以外的环境描述数据不可能很多,因此在建模子系统中可对环境数据作适当的修整。其思路是:保持原有环境数据的真实性,在最后的环数据点处结合上一点向外做三维远程插值以构建虚拟的环境点。这种方法被证明很有效果。

· 光照的处理

为了加速场景的绘制以提供实时的观察能力,可在光照处理上采取措施:

使用缺省的光源。虽然 OpenGL 允许定义多个光源,但每加入一个光源会带来大量的计算工作,最终影响着色性能。

对于公路路面、山体、植被等环境对象,由于其材质特点,不考虑这些对象的复杂材料参数(如材质的镜面反射强度、镜面反射指数、发射光等),仅使用缺省的材质参数,定义其一般的颜色属性即可。

对于建筑物、汽车等对象,可以赋予比以上对象丰富的属性。

4 实时着色/交互控制

在三维空间里,选择好一个有利的观察点,然后才能看到视觉良好的逼真三维图像。对图像进行变换与照相机拍摄物体的过程是相似的。

在 OpenGL 中,二维、三维顶点在内部都当成 4 个坐标组成的三维齐次坐标 (x, y, z, w) ,若 $w > 0$, (x, y, z, w) 和 (ax, ay, az, aw) 代表齐次坐标相同点,而 $(x, y, z, 1)$ 则是三维 Euler 坐标的一点 (x, y, z) 。若 $w < 0$,齐次坐标 (x, y, z, w) 代表 Euler 坐标 $(x/w, y/w, z/w)$, $w = 0$ 则代表在无穷远处的点。在 $w < 0$ 时,OpenGL 处理齐次裁剪平面时有时会弄错,因此应尽量使用 $w > 0$ 的值。

投影变换除了确定观察范围外,还决定物体投影到屏幕的方式:透视或正交,在大多数三维应用里很显然需要的是透视投影变换。

在建模数据处理时,可允许用户设置或系统自动提供观察场景时的视点初始位置及视线并据此生成了一条浏览路径 P,该路径一般由若干关键控制点构成。

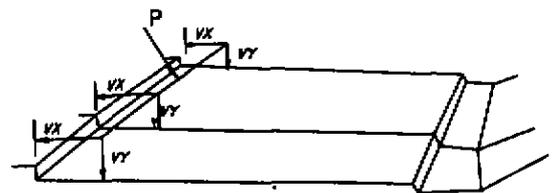


图 3 浏览路径

在实时着色过程中,还应该允许用户在任何时刻移动该路径,提供的移动措施可以包括在着色窗口中使用鼠标和键盘,系统随时记录在着色窗口中纵横方向上的相对偏移量,将其作用于每一个场景:

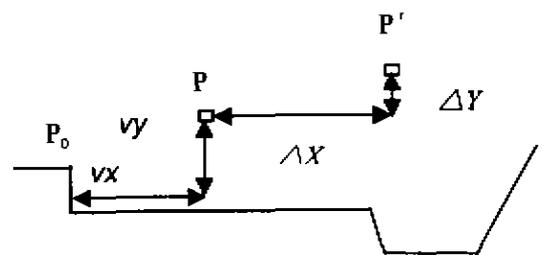


图 4 在每一个场景的观察点移动

着色每一个场景时,根据用户动态移动视点的情况,动态计算出取景位置(即视点位置)。不同的应用,处理情况会有所不同。如对于道路浏览,可预先生成一条行驶(用于浏览)路径,用户的动态移动使这条路径在每一个断面上作偏移,在实现时以少量的插值运算即可得到动态的视点位置。

如此形成的浏览观察点路径是放置视点一虚拟眼睛的位置,视点在前方的着陆点(在 OpenGL 中称为 LookAt Position,这里我们称为 LookAt 点)在哪里呢?

解决方法很多,笔者常用的方法是:

在进入实时着色阶段时,这两点均沿同一条路径 P 移动,即眼睛是平视前方的。与行驶路径 P 可被用户改变为 P' 一样,LookAt 点也可以让用户动态移动,而形成一条新的 LookAt 路径,其方法与行驶路径的动态移动是一样的。

提供了动态移动视点、LookAt 点的功能后,用户使用更加灵活自如,犹如乘坐在一辆直升飞机中沿着设计好的公路、小区、城市等行驶,不仅汽车可方便地改变位置(移动视点),观察的视线(LookAt 点与视点连接而成)也可随时上下左右地移动。

参考文献:

- [1] 孙家广,杨长贵. 计算机图形学[M]. 北京:清华大学出版社,1995.
- [2] 黄伟伦,凌明煌,薛沛宏等. MS-Windows 多媒体程序设计实务与范例[M]. 武汉:华中理工大学出版社,1996
- [3] STAN TRUJILLO. 高性能 Windows 图形程序设计[M]. 北京:机械工业出版社,1998.
- [4] 贾志刚. 精通 OpenGL[M]. 北京:电子工业出版社,1998
- [5] 李薇,徐国标,尹皓等. OpenGL 3D 入门与提高[M]. 成都:西南交大出版社,1998

Developing Real Time 3D Applications with OpenGL

XU Shou-rong

(College of Spaceflight Staff of Southwest, Chongqing 400021, China)

Abstract: 3D-computer graphics technology has progressed to the real time stage. One of the most attracting topics today is how to effectively develop real time 3D systems. The author of this paper has experienced developing some OpenGL based application softwares, and has summarized lots of key techniques including initializing methods, modeling in classes and groups, textures usage, real time viewing control by users and viewing position processing. All these are successfully adapted in the implementation of such systems as real time special effect TV title system and visual road design and evaluation system. In order to be a reference to other researchers and developers, and to make progress together with our competitors, a methodical description is presented here to summarize some typical steps, methods and some valuable experiences.

Key words: 3D engine; texture; bitmap; displayList

(责任编辑 吕蓉英)

·下期论文摘要预告·

三相四线制并联有源电力滤波器

周林,秦梅,易强,叶一麟

(重庆大学电气工程学院,重庆 400044)

摘要: 大量关于有源电力滤波器的研究仅限于三相三线制系统,而没有考虑三相四线制系统。针对不对称三相四线制系统中零序电流的补偿,提出采用三相四线制并联有源电力滤波器来滤除其中的零序电流。详细阐述了这种三相四线制并联有源电力滤波器的主电路结构、工作原理、数学模型和检测方法,并提出了一种不经坐标变换的滞环控制法。仿真结果表明这种有源电力滤波器能有效滤除三相四线制系统中的基波零序、基波负序和谐波等有害电流。

关键词: 有源电力滤波器; 零序补偿; PWM 电流控制器