

基于顶点 Shader 实现的光照计算技术

苏 蕴,陈操宇,郭善良

(上海师范大学 数理信息学院,上海 200234)

摘 要: 真实的光照效果模拟一直是计算机图形学中一个备受瞩目的课题。鉴于此,在 Microsoft Visual Studio 开发环境下,利用 DirectX 9.0 软件包进行 3D 开发,结合顶点 shader 技术编程实现了 Direct3D 支持的三种灯光(平行光、点光、聚光)的顶点光照效果,并与使用标准图形管线实现的光照效果进行了对比。发现前者可以实现更为灵活且真实的光照效果。

关键词: 计算机图形学;顶点光照;shader 程序

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2006)12-0058-03

Computing Technology of Lighting Based on Vertex Shader

SU Yun, CHEN Cao-yu, GUO Shan-liang

(Mathematics and Science College, Shanghai Normal University, Shanghai 200234, China)

Abstract: The realistic representation of lighting effect is one of the key topics in real-time computer graphics. Per-vertex lighting effect of three types of light (directional light, point light and spot light) supported by Direct3D is realized using Microsoft Visual Studio, DirectX 9.0 and shader technique. And it is compared with standard graphics pipelining. Discover that the former can realize more flexible and more realistic lighting effect.

Key words: computer graphics; per-vertex lighting; shader programs

0 引 言

光照用来表示材质和光源之间的相互作用,也可以表示光源与所绘制几何对象之间的相互作用,能够有效地增强物体的真实感。因此真实的光照效果模拟一直是计算机图形学领域中一个备受瞩目的课题。Direct3D 定义了三种不同类型的灯光即平行光、点光和聚光来模拟世界中的光照环境,并通过标准图形管线来实现。然而通过这种方法实现的光照效果限制了使用者对更真实的光照效果的追求,在某些应用场景下,显得非常拘谨和呆板,例如,Direct3D 定义了聚光在内部锥体部分的光照强度保持不变,而在现实中大部分光源(如手电筒等)发出的光线在发光体轴线周围的强度是逐渐增强的。

随着图形处理器技术的迅速发展,新一代的 GPU 具有了可编程特性,那么程序员可以通过对 GPU 编写一段程序来控制渲染输出效果,从而实现一些固定流水线中没有定义的效果,这种程序被称为 shader 程序,它被编译成显卡的 GPU 指令序列在显示芯片上运行^[1]。shader 程序分为两种:顶点 shader 和像素 shader。其中通过顶点 shader,可以提供更加灵活的顶点光照模型计算方法。

1 影响光照效果的因素

光照效果是由环境光分量、漫反射光分量和镜面反射光分量共同决定的。环境光是经过环境多次散射,无法确定其方向的光。在渲染场景时,必须加上一定量的环境光,否则,将无法看到场景中的任何物体。散射光来自同一个方向,到达粗糙物体表面后,将沿着各个方向均匀地散射。镜面反射光则来自于特定的方向,经光滑物体表面反射到特定的方向。总的光照强度 I 就是环境光分量强度、漫反射光分量强度和镜面反射光分量强度之和,即:

$$I = I_{\text{Ambient}} + I_{\text{Diffuse}} + I_{\text{Specular}} \quad (1)$$

对于照射到特定物体的光线来说,强度越大,光线就越容易被肉眼所感知。光线的强度由三个分量来描述,分别是环境光分量、漫反射光分量和镜面反射光分量。然而并不是所有由光源发出的光线都能到达人的眼睛,光源发出的光线一部分被物体吸收,剩下的部分被物体反射,这些反射光到达人的眼睛,才能被眼睛的光感细胞所感应^[2]。因此物体表面的材质特征也会影响光照效果,材质也有环境光分量、漫反射光分量和镜面反射光分量,它们决定了材质对光线相应分量的反射率。材质的另外一个描述分量是自发光分量,用以模拟物体自身发射的光。物体表面顶点的法向量确定有多少来自光源的光将照射到顶点上,当光线垂直地照射到物体表面时,总比斜照时要亮一些。

另外在光线发生衰减的情况下,光源和顶点的相对位

收稿日期:2006-03-29

作者简介: 苏 蕴(1982-),女,山东菏泽人,硕士研究生,主要研究方向为计算机图形学;陈操宇,教授,博士,主要研究方向为虚拟现实和数据库等;郭善良,教授,博士;主要研究方向为计算机图形学和数据库等。

置也会对光照效果产生影响。

2 光照效果的顶点 shader 程序实现

在硬件支持的情况下,顶点 shader 代替了标准图形管线中的坐标变换和光照变换阶段。在顶点 shader 中进行光照计算的步骤是:首先给定面片的各顶点信息,包括法线向量、位置等;再对各顶点采用适当的光照模型进行光照计算,求出各顶点的颜色值;最后通过线性插值求出该面片上各个像素的颜色值^[3]。基于顶点 shader 的光照计算绘制流程如图 1 所示。

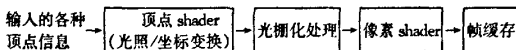


图 1 基于顶点 shader 的光照计算绘制流程

DirectX 9 通过 effects framework 把与一个渲染效果相关的任务整合到一个效果文件(.fx 文件)中,效果文件中可以集成多个不同的渲染效果,不同的三维场景可以根据不同的需要来调用不同的效果。这样不用重新编译应用程序源码就能改变一个效果的执行,并使得更新效果的过程变得更轻松。对于顶点 shader 或像素 shader 程序,一般包括:全局变量(Global variables,这些变量将从主函数中传过来)、输入/出结构体和程序入口函数。本程序主要使用顶点 shader 实现了上述三种灯光照射墙壁的基本效果。

2.1 全局变量的设置

在主函数中把表 1 中影响光照效果的分量通过全局变量传入 shader 程序。这里假定墙壁是比较粗糙的,忽略了墙壁的镜面反射。

表 1 全局变量名称及含义

变量名称	含义	变量名称	含义
$M_{Ambient}$	材质的环境光分量	$Attenuation2$	聚光的二次衰减系数
$M_{Diffuse}$	材质的漫反射光分量	A_{glob}	全局环境光分量
$M_{Emissive}$	材质的自发光分量	Type	光源的类型
$S_{Ambient}$	光线的环境光分量	Falloff	聚光的内外锥衰减
$S_{Diffuse}$	光线的漫反射光分量	Phi	聚光的外部锥体角度
$S_{Specular}$	光线的镜面反射光分量	Theta	聚光的内部锥体角度
$S_{position}$	视点变换后光源的位置	M_{proj}	投影变换矩阵
$S_{Direction}$	视点变换后光源的单位化方向向量	M_{camera}	视点变换矩阵
$Attenuation0$	聚光的常量衰减系数	M_{normal}	法线变换矩阵
$Attenuation1$	聚光的线性衰减系数		

利用单个矩阵可以对点、线、多边形和其他几何图形进行一致性的变换。除此之外,还可以对线或者多边形上的向量进行变换,但是却无法对表面法线进行变换^[4]。几何图形沿着非法线方向的缩放往往会导致法线的不正确的变换。因此,法线必须通过用来变换几何图形的逆矩阵的转置矩阵进行变换^[5]。在此处 shader 程序中,光源和顶点的相对位置均是在视点空间中进行计算的,故法线变换矩阵 M_{normal} 是通过对视点矩阵求逆并转置获得的。

2.2 输入输出结构体

顶点信息通过输入结构 VS_INPUT 来描述,包括顶点的位置和法向量。输出结构 VS_OUTPUT 描述了经

过顶点光照计算后顶点的位置和颜色信息。

```
struct VS_INPUT
{
    Vector position: POSITION; //顶点在世界坐标系中的位置
    向量
    Vector normal: NORMAL; //顶点在世界坐标系中的单位
    法线向量
};
struct VS_OUTPUT
{
    Vector position: POSITION; //顶点在投影坐标系中的位置
    向量
    Vector diffuse: COLOR; //顶点颜色
};
```

2.3 三种灯光的顶点光照计算方法

通过顶点光照的计算,需要最终输出经过投影变换的顶点的位置,以及顶点的颜色。对于顶点位置的计算,只要将输入的世界坐标系中的顶点位置,依次乘以视点变换矩阵和投影变换矩阵即可。顶点颜色的计算,按照灯光的类型将具体讨论如下。

2.3.1 平行光的光照方程

平行光表示在场景的无穷远处生效的光,它只有颜色和方向,没有发光的位置,其光线互相平行,它可用来模拟太阳光等不随距离减弱的自然光,因此计算量较小,在场景中大量使用这种光源能够有效地提高程序的运行效率。

环境光分量强度由全局环境光 A_{glob} 和光源中的环境光分量以及材质中的环境光分量共同决定的,即:

$$I_{Ambient} = M_{Emissive} + M_{Ambient} \odot A_{glob} + S_{Ambient} \quad (2)$$

其中,符号 \cdot 表示向量之间的点乘, \odot 表示分量相乘。

入射光线的单位方向向量跟顶点单位法线向量的夹角小于 90 度时才有可能产生反射光线,即:

$$I_{Diffuse} = \max(N \cdot -S_{Direction}, 0) * S_{Diffuse} \odot M_{Diffuse} \quad (3)$$

其中, N 表示顶点的单位法线向量,在本程序中, $N = \text{mul}(\text{input.normal}, M_{normal})$, 符号 $-$ 表示对向量方向取反。

将方程(2),(3)代入(1)就得到平行光的顶点光照计算方程:

$$I = M_{Emissive} + M_{Ambient} \odot A_{glob} + S_{Ambient} + \max(N \cdot -S_{Direction}, 0) * S_{Diffuse} \odot M_{Diffuse} \quad (4)$$

2.3.2 点光的光照方程

点光源具有特定的位置和颜色亮度,能够向四面八方发出光线。点光的光照方程和平行光类似,但由于点光的强度会随着顶点距离光源的距离远近而发生衰减,因而需要乘以距离衰减因子。

$$I = M_{Emissive} + M_{Ambient} \odot A_{glob} + (S_{Ambient} + \max(N \cdot P, 0) * S_{Diffuse} \odot M_{Diffuse}) * (Attenuation0 + Attenuation1 * d + Attenuation2 * d^2)^{-1} \quad (5)$$

其中, P 表示由光照顶点指向光源的单位方向向量,即 $P = \text{normalize}(V_{position} - S_{position})$; $V_{position}$ 表示光照顶点的位置向量,在本程序中 $V_{position} = \text{mul}(\text{input.position},$

M_{camera}), d 为物体的顶点到光源的距离。

2.3.3 聚光的光照方程

聚光是三种光源中计算量最大的一种光源。一方面它同样会随着物体的远近而发生衰减;另一方面,还应判断聚光源到顶点的向量在其发光锥体的哪一部分。即计算在内外锥体上的衰减因子 SpotFactor, 设变量 $Rho = -P \cdot S_{\text{Direction}}$, 则:

$$\text{SpotFactor} = \left(\frac{\max(Rho - \cos(\Phi/2), 0)}{\cos(\Theta/2) - \cos(\Phi/2)} \right)^{\text{Falloff}} \quad (6)$$

由方程(5),(6)得到的聚光的光照方程为:

$$I = M_{\text{Emissive}} + M_{\text{Ambient}} \odot A_{\text{glob}} + (S_{\text{Ambient}} + \max(N \cdot P, 0) * S_{\text{Diffuse}} \odot M_{\text{Diffuse}}) * (\text{Attenuation0} + \text{Attenuation1} * d + \text{Attenuation2} * d^2)^{-1} * \left(\frac{\max(Rho - \cos(\Phi/2), 0)}{\cos(\Theta/2) - \cos(\Phi/2)} \right)^{\text{Falloff}} \quad (7)$$

通过使用方程(7)计算的光照效果使光线在发光锥体轴线上的亮度明显增强,更加接近现实世界。

如果一个顶点受多光源的照射,那么最终的颜色值 I_{tot} 应该是多个光源光照效果的叠加。

$$I_{\text{tot}} = \sum_{i=1}^n I_i$$

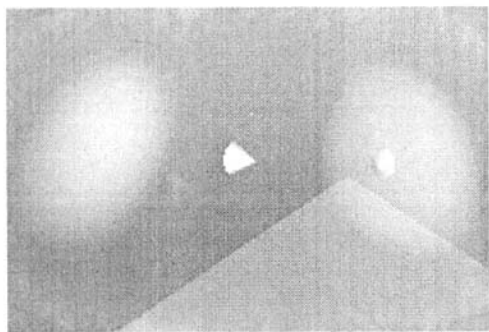


图2 shader 程序实现的聚光光照效果

3 总结与展望

通过顶点 shader 程序可以比较灵活地实现更为真实的光照效果。图2、图3对比了运用标准图形管线和 shader 程序实现的聚光的光照效果的不同。在图2中,靠近发光锥体轴线的部分,光线强度是逐渐增强的,这种效果更加贴近现实世界。但是还存在需要改进的地方,比如,墙壁较光滑时,还需要考虑它的镜面反射光。笔者将在以后的研究中做进一步的改进。

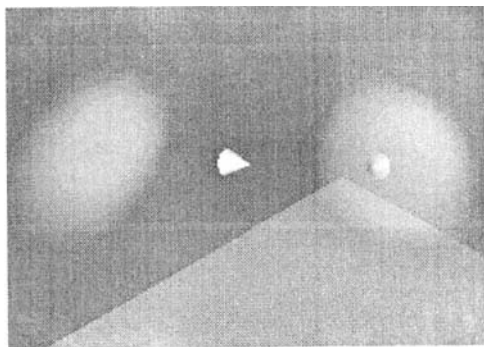


图3 标准图形管线实现的聚光光照效果

参考文献:

- [1] Fernando R. GPU 精粹——实时图形编程的技术、技巧和技艺[M]. 北京:人民邮电出版社, 2006:100-106.
- [2] 陈卡. DirectX9 3D 图形程序设计[M]. 上海:上海科学技术出版社, 2003:92-100.
- [3] 朱腾辉, 刘学慧, 吴恩华. 基于像素的光照计算技术[J]. 计算机辅助设计与图形学学报, 2002, 14(9): 861-865.
- [4] Akenine-Moller T, Haines E. 实时计算机图形学[M]. 北京:北京大学出版社, 2004:20-21.
- [5] Hanrahan P. An Introduction to Ray Tracing[M]. London: Academic Press Inc, 1989:79-120.

(上接第 57 页)

- [3] Schapire R E. The strength of weak learnability[J]. Machine Learning, 1990(2): 197-227.
- [4] Zhou ZH, Wu J, Tang W. Ensembling neural networks: Many could be better than all. Artificial Intelligence, 2002, 137(1-2): 239-263.
- [5] 吴启迪, 汪镭. 智能微粒群算法的研究及应用[M]. 南京: 江苏教育出版社, 2005.
- [6] Kennedy J, Eberhart R C. Swarm intelligence[M]. San Francisco: Morgan Kaufmann, 2001.
- [7] 康琦. 微粒群优化算法的研究与应用[D]. 上海: 同济大学, 2005.
- [8] 吴启迪, 汪镭. 智能蚁群算法及应用[M]. 上海: 上海科技教育出版社, 2004.
- [9] Dorigo M, Caro G D. Ant colony optimization: a new meta-heuristic[C]//In: Proc. of the 1999 Congress on Evolutionary Computation, Vol 2. Washington: IEEE Press, 1999: 1470-1477.
- [10] Dorigo M, Caro G D, Gambardella L M. Ant algorithms for discrete optimization[J]. Artificial Life, 1999, 5(2): 137-172.
- [11] Dorigo M, Maniezzo V, Colnani A. The ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 29-41.
- [12] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [13] 燕忠, 袁春伟. 增强型的蚁群优化算法[J]. 计算机工程与应用, 2003, 39(23): 62-64.
- [14] Efron B, Tibshirani R. An Introduction to the Bootstrap[M]. New York: Chapman & Hall, 1993.

基于顶点Shader实现的光照计算技术

作者: [苏蕴](#), [陈操宇](#), [郭善良](#), [SU Yun](#), [CHEN Cao-yu](#), [GUO Shan-liang](#)
作者单位: [上海师范大学, 数理信息学院, 上海, 200234](#)
刊名: [计算机技术与发展](#) 
英文刊名: [COMPUTER TECHNOLOGY AND DEVELOPMENT](#)
年, 卷(期): 2006, 16(12)
被引用次数: 1次

参考文献(5条)

1. [Femando R GPU精粹—实时图形编程的技术、技巧和技艺](#) 2006
2. [陈卡 DirectX9 3D图形程序设计](#) 2003
3. [朱腾辉, 刘学慧, 吴恩华 基于像素的光照计算技术\[期刊论文\]-计算机辅助设计与图形学学报](#) 2002(09)
4. [Akenine-Moller T, Haines E, 曹建涛 实时计算机图形学](#) 2004
5. [Hanrahan P AnIntroduction to Ray Tracing](#) 1989

相似文献(4条)

1. 期刊论文 [苏蕴, SU Yun 利用Vertex Shader技术实现真实的光照效果 -科技信息2010\(27\)](#)
Vertex Shader代替了标准图形管线中的坐标变换和光照变换阶段, 能够实现更为灵活且真实的光照效果. 本文使用结合Vertex Shader技术编程实现了平行光、点光、聚光的顶点光照效果, 并与使用标准图形管线实现的光照效果进行了对比.
2. 学位论文 [沈潇 三维场景实时阴影算法的研究与实现](#) 2006
作为能增加虚拟场景真实感的关键技术——光照和阴影实时渲染技术, 在三维游戏、虚拟训练系统、三维地理信息系统等方面都有广泛的应用. 然而, 针对体现高真实感的软阴影算法, 虽然在近3年里提出的算法众多, 但是还是存在不少问题, 比如无法形成自阴影、需要区分生成阴影的遮挡体和接受体, 忽略光源的大小和形状、只考虑半影的内半影或外半影的渲染等等. 本文文基于现代图形处理器的可编程特性, 在吸取三维计算机图形学、计算几何、科学计算可视化、虚拟现实的先进理论和技术成果的基础上, 对实时光照阴影技术进行了具体而系统的研究. 本文文的主要研究工作有:
(1) 利用可编程图形硬件可编程特性, 实现了基于Phong 着色模型的逐像素着色光照技术及基于Gouraud 着色模型的逐顶点光照技术, 这种实现方式与传统的固定管道实现方式相比, 具有灵活性高、真实感强的优点. 另外虽然固定管道已经实现了Gouraud着色模型, 但采用可编程图形硬件的纯顶点着色器实现起来速度更快.
(2) 分析了目前流行的硬阴影生成算法的实现原理, 比较各算法的局限性和优越性, 并且针对阴影图算法中由于算法本身所带来的偏移量问题, 提出了自适应偏移量算法. 该算法只需设置简单的几个参数, 由程序自动分析场景特性取得一个合理的偏移量, 这种方法相对于目前存在的解决方案具有不再需要由程序员依据场景手动调整偏移量大小的优点.
(3) 在分析已有的软阴影算法基础上, 提出一种基于内外半影的软阴影算法, 该算法扩展了chan的光滑翼算法, 在物体轮廓边处构成内外半影翼, 并利用了第二层向光面深度值控制内半影的生成, 整个算法结合了阴影体算法及阴影图算法的特点, 并且可根据光源大小自动调整半影大小. 本文使用可编程图形硬件实现该算法, 实现过程中充分利用图形硬件的并行处理加速的特性, 使算法满足了实时性要求.
(4) 最后, 本文从渲染效果, 渲染速度及系统运行的要求几方面与传统算法进行了分析比较. 算法渲染效果比chan的光滑翼算法效果更接近于光线跟踪算法实现的物理精确效果, 尤其在重叠场景中能真实反映光源、接受体及遮挡体的空间几何特征. 由于算法需要计算场景轮廓边, 所以渲染速度会受到场景复杂程度的影响. 此外, 算法中需要取得第二层向光面深度值, 因此算法的运行依赖于可编程图形硬件, 适应了现代图形硬件的发展趋势.
3. 学位论文 [袁亚杰 基于可编程图形硬件的实时图形技术研究](#) 2006
计算机实时图形技术是计算机三维游戏, 计算机辅助设计与制造(CAD/CAM), 数字媒体创作(DCC)、虚拟现实等领域中的基本问题. 最近几年, 随着这些应用领域的飞速发展, 人们对实时图形的要求也越来越高, 提出了更为复杂的, 能逼真再现真实场景的实时渲染要求. 为了满足这些应用领域的要求, 基于图形处理器(GraphicProcessingUnit)的实时渲染技术成为了计算机图形学的一个重要的研究方向.
文章主要研究了基于最新GPU的实时渲染技术, 主要涉及以下方面: 逐像素光照计算、实时阴影、材质与丝绸材质的渲染及蒙皮网格体的渲染. 此外, 文章还结合作者将来的工作, 简要阐述了基于硬件优化的基本原则. 考虑到实时图形技术近几年一的飞速发展, 文章涉及到的图形处理器都是支持ShaderModel2.0及以上版本的. 上述均体现文章紧密结合日新月异实时图形渲染技术, 跟上时代前进的步伐. 以下是文章各章节的内容简介:
第1章绪论. 主要介绍实时图形技术的重要性、国内、外研究的现状、课题研究的的意义以及论文涉及的关键技术.
第2章系统开发的核心技术. 主要介绍计算机实时图形及软硬件平台相关的技术. 其中首先介绍了计算机图形学及三维向量代数的基础知识, 随后介绍了图形硬件和图形应用编程接口(API)的发展, 并从应用角度分析了图形软硬件的发展趋势.
第3章系统开发的详细设计. 首先介绍了逐顶点光照和逐像素光照的区别, 以及逐像素光照在本系统中的实现, 然后介绍了几种经典的阴影生成方式以及它们的改进和实现, 提出基于shadowmapping的软阴影, 改进基于GPU的shadowvolume技术. 随后是在本系统中实现的基于光学的玻璃与基于BRDF理论的丝绸材质渲染技术, 最后介绍了蒙皮网格体动画的渲染技术, 并使用GPU加速技术提高了性能. 文章还结合具体实现细节提出了一些基于图形硬件的渲染优化技术.
第4章研究工作总结. 对以上各章介绍的技术、方法进行了总结, 指出其中的创新点及有待改进的地方.
4. 学位论文 [付恺 基于可编程图形硬件的实时图形技术研究](#) 2005
文章所介绍的技术都是围绕Shader Model 2.0展开. 以下是文章各章节的内容简介: 第1章绪论. 主要介绍实时图形技术的重要性、国内、外研究的现状、课题的目的和意义以及论文涉及的关键技术. 第2章系统开发的核心技术. 主要介绍计算机实时图形及软硬件平台相关的技术. 其中首先介绍了计算机图形学及三维向量代数的基础知识, 随后介绍了图形硬件和图形应用编程接口(API)的发展, 并从应用角度分析了图形软硬件的发展趋势. 第3章系统开发的详细设计. 主要讨论大规模复杂光照情况下的实时渲染技术. 文章基于现有的研究成果并结合目前先进的图形硬件, 提出了几种新的实时图形渲染技术. 首先介绍了逐顶点光照和逐像素光照的区别, 以及逐像素光照在本系统中的实现, 然后介绍了几种经典的阴影生成方式以及它们的改进和实现. 随后是在本系统中实现的基于费曼定理的室内水面渲染技术, 最后介绍了蒙皮网格体的渲染技术. 在介绍这些技术的同时, 文章还结合具体实现细节提出了一些基于图形硬件的渲染优化技术. 第4章研究工作总结. 对以上各章介绍的技术、方法进行了总结, 指出其中的创新点及有待改进的地方.

引证文献(1条)

1. [程利利](#), [姜晓彤](#), [王庆利](#), [赵正旭](#) [基于GLSL的真实感光照算法研究与实现](#)[期刊论文]-[现代电子技术](#) 2009(18)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjtz200612019.aspx

授权使用: 复旦大学图书馆(fddxlwxsjc), 授权号: d26b1e1c-34b1-4839-9514-9ea00117815e

下载时间: 2011年3月8日