

基于 HLSL 的立方体映射技术研究 with 实现

杜昊, 李光耀

(同济大学 CAD 研究中心, 上海 201804)

摘要: 计算机仿真领域中, 在满足实时性的基础上增强仿真的真实感一直是一个重要的问题。立方体映射技术能够使两者达到一个较好的平衡效果。文中对立方体映射技术的原理进行了详细的研究, 并通过 HLSL 语言在 Direct 3D 的环境下实现。并且讨论了立方体映射技术的两种改进方法, 一种加入了折射效果, 一种使用动态地更新立方体纹理的方法。两种方法增强了立方体映射技术绘制下场景的真实感, 拓宽了立方体映射技术的适用范围。

关键词: 立方体映射; 高级着色器语言; 环境映射

中图分类号: TP391.9

文献标识码: A

文章编号: 1673-629X(2011)11-0004-04

Research and Implementation of Cube Mapping Technique Based on HLSL

DU Hao, LI Guang-yao

(CAD Research Center, Tongji University, Shanghai 201804, China)

Abstract: It's an important problem to strengthen the reality of simulation on the base of real-time in the field of computer simulation, and cube mapping technique can make them reach the balance. It introduces the principle of cube mapping technique in detail, which is programmed through HLSL language in Direct 3D environment. And it discusses two improving methods of cube mapping technique, adding refractive effects and updating cube texture dynamically. The two methods strengthen the reality of scene which is rendered through cube mapping technique, and expand the application range of cube mapping technique.

Key words: cube mapping; HLSL; environment mapping

0 引言

高级着色器语言 (High Level Shader Language, 简称 HLSL), 是由微软拥有及开发的一种语言, HLSL 独立地工作在 Windows 平台上, 只能供微软的 Direct 3D 使用^[1,2]。HLSL 的主要作用为将一些复杂的图像处理, 快速而又有效率地在显示卡上完成, 与组合式或低阶 Shader Language 相比, 能降低在编写复杂特殊效果时所发生编程错误的机会。HLSL 现已经整合到了 DirectX SDK 中^[3]。

立方体映射是环境映射中的一种常用的方法。环境映射在计算机图形学领域中是用预先计算的纹理图像模拟复杂镜面的一种高效的方法^[4-6]。纹理用来储存被渲染物体周围环境的图像。环境映射最常用的有

两种方法, 一种是球面映射, 在这种方法中周围环境在一个镜面球体上反射得到单一纹理图像。球形映射有一些明显的限制, 其中之一是, 由于纹理属性的原因, 在球形映射物体的后面有一个突变点。目前球形环境映射在当今的图形应用中几乎已经销声匿迹了。另外一种常用方法是立方环境映射^[7], 在这种方法中环境在立方体的六个表面上显现出来, 所以需要保存为六个正方形的纹理。这种实现方法比传统的光线跟踪算法效率更高, 但是需要注意的是这种方法是实际反射的一种近似, 有时甚至是非常粗糙的近似。

1 立方体映射介绍

从概念上讲, 立方体贴图就是用立方体的 6 个面来表现环境, 这个立方体用来包裹场景中的一个或多个物体^[7]。一般把观察点固定在物体中心来接受环境图, 绘制了六个视图。考虑将观察点固定在房间中心。如果房间是空的, 则视图将包括四面墙、天花板和地板。立方体映射的问题之一是如果把反射光束等价地看作是由多边形顶点处反射的观察向量形成的, 则光束可以索引多个图。在这种情况下, 可以将多边

收稿日期: 2011-05-03; 修回日期: 2011-08-10

基金项目: 国家 863 计划重点项目 (2010AA122200); 上海市科委国际合作项目 (10510712500)

作者简介: 杜昊 (1986-), 男, 山东潍坊人, 硕士研究生, 主要研究领域为计算机仿真; 李光耀, 教授, 博士生导师, 主要研究领域为计算机辅助设计分析与仿真、城市仿真与城市规划设计、图形图像技术、虚拟现实等。

形细分,使每一个小块都被限定到一个映射图中,如图 1 所示^[8]。

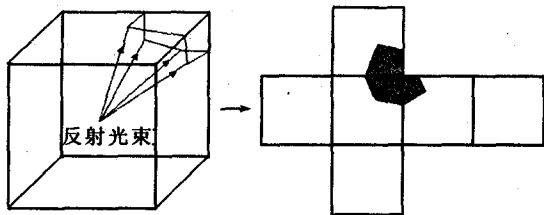


图 1 反射光束可覆盖多个映射图

对于立方体映射图,需要确定一种能从三维观察向量向一个或多个二维映射图的映射算法。假如考虑反射观察向量与环境图立方体处于相同的坐标系框架中(如果观察是通过把虚拟的或真实的摄像机沿着世界坐标系的轴在两个方向放置来建立的),则映射规则如下:

对于一个标准化的反射向量:

1) 求出它相交的小平面——映射因数。这是一个将单位化的反射观察向量与以原点为中心的单位立方体的大小进行简单的比较。

2) 把这些分量映射到 (u, v) 坐标上。例如,与负的 Z 轴的表面法向相交的点 (x, y, z) 由下式给出:

$$u = x + 0.5$$

$$v = -z + 0.5$$

规范如图 2 所示^[8,9]。

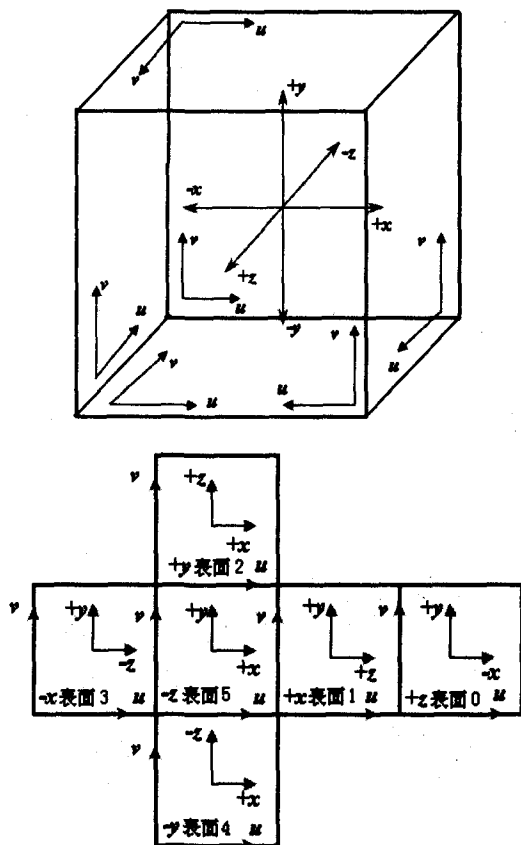


图 2 立方体映射图规范

图 3 显示了一个立方体映射的例子^[10]。环境中的茶壶分别是用简单的六色图和它自身的层次环境来进行立方体映射的。从这里便能清楚地看到茶壶身上的环境是贴图形成的。在实际应用中,立方体贴图存储时不是像图中那样呈“十”字形存在,而是从左至右按“ $+x$ 面、 $-x$ 面、 $+y$ 面、 $-y$ 面、 $+z$ 面、 $-z$ 面”的顺序排成的“一”字形的贴图^[5,11]。



(a) 映射图由 6 种简单的颜色组成



(b) 映射图由周围环境的贴图组成

图 3 不同的贴图产生不同的效果

该示例是立方体映射最简单的表现形式,但是它已经可以产生很漂亮的结果了。由于要使用立方体环境映射,因此首先需要定义一个立方体纹理采样器 CubeMapSampler。继而在顶点渲染器中,计算了入射向量和反射向量,并且将反射向量传递给像素渲染器。然后像素渲染器从立方体环境纹理中取出颜色值,并将其作为输出值使用。顶点渲染器和像素渲染器主要代码如下:

```
Out. Pos = mul( Pos, matWorldViewProj ); //坐标变换
```

```
float3 Norm = normalize( mul( Normal, matWorld )); //计算世界空间中的顶点法线并标准化
```

```
float3 PosWorld = normalize( mul( Pos, matWorld ));
```

```
float3 Incident = normalize( PosWorld - vecEye ); //计算从顶点指向观察点方向向量并标准化
```

```
Out. Reflect = normalize( reflect( Incident, Norm )); //计算立方体映射的反射向量
```

```
return texCUBE( CubeMapSampler, In. Reflect ); //像素渲染器
```

在顶点渲染函数中,顶点位置 Pos 通过应用程序提供的世界-观察-投影复合变换矩阵 matWorldViewProj 完成坐标变换,法向量 Normal 则通过世界矩阵 matWorld 变换到世界坐标系下,入射光向量 Incident 由世界空间中顶点的位置 PosWorld 减去摄影机/眼睛

位置 vecEye 而得到。归一化的反射向量 Reflect 则借助于 HLSL 编译器所提供的函数 $\text{reflect}()$ 和 $\text{normalize}()$ 计算得到。

在接下来的像素渲染函数中,直接通过立方体纹理采样器 CubeMapSampler ,调用函数 $\text{texCUBE}()$ 从立方体贴图 CubeMap 中采样得到当前像素的颜色值。函数 $\text{texCUBE}()$ 的第一个参数是所使用的立方体纹理采样器,第二个参数是一个 3D 查找向量,这里需要将它指定为顶点渲染函数中计算出的反射向量 In. Reflect 。

2 立方体映射的几种改进

2.1 折射/反射立方体映射

在现实世界中,当光线穿过一种介质到达另一种介质时,会在这两种介质的相接处发生折射。折射效果近似于反射,它重用了反射方程式和一个经过缩放的法向量。将法向量缩短后会导致一个弯曲的折射向量,它看上去就像是摄影机/眼睛向量的延长。

如果法向量的长度缩短至 0,折射向量将与向量 I 相同。如果法向量稍长一些,折射向量相对于表面就弯曲的更多,如图 4 所示^[12]。

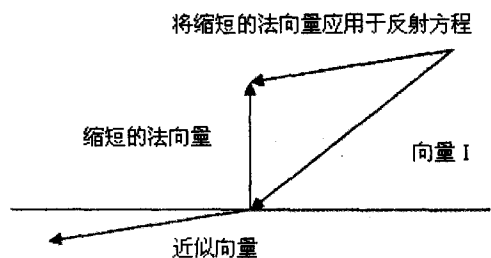


图 4 折射向量

在这段程序中,只需在顶点渲染器主函数中加入折射向量的代码,在像素渲染器主函数中加入折射向量的返回。

```
float3 ShortNorm = mul(Norm, 0.4);
Out. Reflect2 = normalize( reflect( Incident, Short-
Norm)); //折射向量
//像素渲染器主函数
float4 tex1 = texCUBE( CubeMapSampler, In.
Reflect);
float4 tex2 = texCUBE( CubeMapSampler, In.
Reflect2);
return tex2 * 0.5 + tex1;
```

在顶点渲染函数中,使用常量 0.4 将法向量缩短后,存放在变量 ShortNorm 中,然后将这个向量传递给 HLSL 内部函数 $\text{reflect}()$ 来计算折射向量 Reflect2 。如果使用一个较大的常量值(接近于 1.0)会使结果看上去更接近于反射,而使用较小的值(接近于 0)则会使

结果更接近于向量 I ,由此可产生一种缩放的效果。

反射向量 Reflect 和折射向量 Reflect2 随后被传递到像素渲染函数中,像素渲染函数分两次读取立方体贴图。然后,将通过折射向量读取到的立方体贴图的颜色值乘以 0.5,以此来降低颜色的亮度。最后将这个值与通过反射向量读取到的颜色值相加得到当前像素的颜色值。程序运行得到的效果图如图 5 所示。

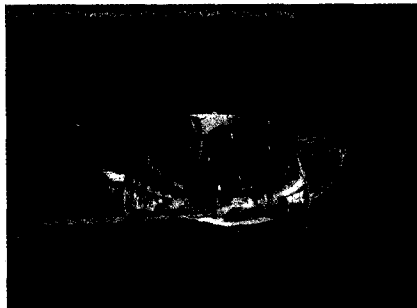


图 5 折射/反射立方体映射效果图

2.2 动态折射/反射立方体映射

之前两种示例,立方体贴图都是预先准备好的,其实立方体贴图的更新过程也可以是动态进行的,即立方体贴图的内容根据环境每一帧都在变化。为了突出这种动态更新的效果,在环境中加入了一架飞机围绕着茶壶飞行来突出这种效果。茶壶上的反射是通过将立方体贴图投影到茶壶上从而实现的。实现动态折射/反射环境映射的关键是如何动态实时地生成立方

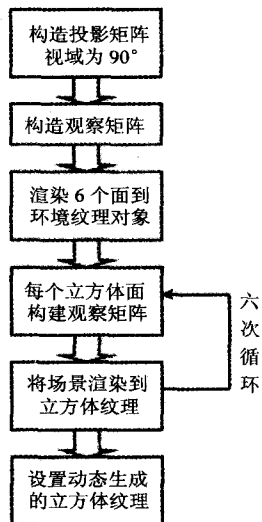


图 6 动态折射/反射立方体映射流程图和效果图

体贴图,其实现流程和程序运行结果如图 6 所示。

然而对于某些硬件设备来说,动态运用立方体映射代价仍然太高,因为 6 个视图必须对物体当前位置每一帧进行渲染。因此可以每五帧或在物体移动一定距离之后生成一次立方体贴图来减小这个代价^[8]。

3 结束语

文中对立方体映射技术进行了研究,并使用 HLSL 语言将其实现。立方体映射的实现较为简单,但缺点也非常明显:一个物体只能反射环境而不能反射其自身;场景中需要映射的每个物体需要单独有一张图;只有物体相对于环境来说较小时这一方法(在几何学上)才是正确的等等^[8]。但是它是一种高效的计算反射的技术,这对于绘制那些反射其所处空间中的环境的闪光物体是一种捷径,这一方法对于镜面反射几乎可以达到光线跟踪程序的质量。因此在实时性要求比较高或无法进行大量计算的情况下,使用立方体映射是一种合适的选择。

参考文献:

- [1] 王弟伟,沈春林. 基于 Direct3D 的 HLSL 高层着色语言实现着色效果[J]. 计算机应用研究,2006,23(11):241-

(上接第 3 页)

入各机器的平均生产率、故障率和修复率,以及已知和未知条件,然后系统根据输入,调用 DDX 算法,最后将结果返回给用户,结果证明是正确的,算法是在 Net 平台下,使用 Visual Studio 2008 开发环境,采用 C# 语言编写。

4 结束语

通过对制造系统的现有知识的表示和管理,再和目前流行的 ASP.NET 技术结合起来,可以使越来越多的设计者或者管理人员来使用这些知识,即解决制造系统中问题的办法,这样对提高企业的生产效率和给管理人员对整个企业生产的管理都带来益处。文中提供的是一种思路和方法,并通过实例证明是有效的,随着制造系统的知识越来越多,要将每条知识添加到所设计的系统中,也需要很大的工作量。

参考文献:

- [1] Choong Y F, Gershwin S B. A Decomposition Method for the Approximate Evaluation of Capacitated Transfer Lines with Unreliable Machines and Random Processing Times[J]. IIE Transactions,1987,19(2):150-159.
- [2] Dallery Y, Gershwin S B. Manufacturing flow line systems: a review of models and analytical results [J]. Queueing Sys-

243.

- [2] 万 薇. 基于 DirectX 的纹理映射技术[J]. 科技广场,2005,10(3):45-47.
- [3] 郝进亮,陈 蕾,娄高鸣,等. 基于 DirectX 的云模拟研究[J]. 计算机技术与发展,2009,19(11):195-197.
- [4] 卢章平,丁立军,戴立玲. 基于分类的纹理映射方法综述[J]. 江苏大学学报(自然科学版),2006,27(5):13-16.
- [5] Grimm C M, Niebruegge B. Continuous Cube Mapping[J]. Journal of Graphics Tools,2007,12(4):25-34.
- [6] Ashikhmin M, Ghosh A. Simple Blurry Reflections with Environment Maps[J]. Journal of Graphics Tools,2002,7(4):3-8.
- [7] NVIDIA Cube Map OpenGL Tutorial [EB/OL]. 1999. <http://developer.nvidia.com/content/cube-map-ogl-tutorial>.
- [8] Watt A. 3D 计算机图形学[M]. 北京:机械工业出版社,2005:191-192.
- [9] 郑根让. DirectX 3D 中的环境映射及特殊效果实现[J]. 宝鸡文理学院学报(自然科学版),2009,29(2):72-74.
- [10] Watt A, Policarpo F. 3D 游戏(卷 2):动画与高级实时渲染技术[M]. 北京:机械工业出版社,2005:170-171.
- [11] 万 亮,黄田津,梁志成,等立方体球面映射[J]. 计算机辅助设计与图形学学报,2008,20(8):978-985.
- [12] 王德才,杨关胜,孙玉萍. 精通 DirectX 3D 图形与动画程序设计[M]. 北京:人民邮电出版社,2007:498-505.

tems,1992,12:3-94.

- [3] Christy D P. Using Computer Simulation to Optimize Flexible Manufacturing System Ddsign[C]// Proceedings of the 1989 Winter Simulation Conference. [s. l.]:[s. n.],1989:396-405.
- [4] Wildermuth S, Wightman J. ADO.NET 应用程序开发[M]. 北京:清华大学出版社,2010.
- [5] 许 云,樊孝忠. 在专家系统中利用关系数据库来表达知识[J]. 计算机工程和应用,2003,22:91-94.
- [6] 邱李华,李晓黎. SQL server 2000 数据库应用教程[M]. 北京:人民邮电出版社,2007.
- [7] 黄胜根,陈蜀宇. 基于 ASP.NET MVC 框架的干教系统的设计与实现[J]. 计算机技术与发展,2010,20(2):190-193.
- [8] 郭秀娟,王春光. 基于 B/S 模式的毕业设计管理系统的开发与实现[J]. 计算机技术与发展,2010,20(3):239-241.
- [9] 吴桂珍,吴曦辉. 域管理模式下的 Web Service 协同反馈系统[J]. 华侨大学学报,2008,29(3):379-382.
- [10] 任兴甜,黄小红,马 严,等. 基于 Web Service 服务管理系统的研究[J]. 计算机工程,2007(4):91-93.
- [11] 王丽侠,楼玉萍,吕君可. 基于 Web 服务的电力信息集成系统[J]. 计算机技术与发展,2009,19(5):173-175.
- [12] 孙红梅,赵 靓. 基于 .NET 的高校教师教学质量评估系统[J]. 计算机技术与发展,2008,18(8):229-231.