

实时仿真中的逼真虚拟地球实现

王 成 戴树岭

(北京航空航天大学 自动化科学与电气工程学院, 北京 100191)

摘 要: 针对虚拟现实真实感不强的问题,提出了一种具有高精度地表纹理的虚拟地球实现方法,构建了一个地表纹理像素精度高达 21 600 像素 \times 10 800 像素、地球表面轮廓凹凸效果和大气层光晕效果的虚拟地球.采用几何计算来构建地球形状模型,将地球按经纬度分成 128 块,分别计算每块区域顶点坐标、法向量、纹理坐标;采用混合纹理技术对地表纹理、云层纹理及云层法向量图进行 3 层纹理混合实现地球表面的云层和轮廓凹凸效果;采用实时移动与放缩带大气层纹理的正方形面板来实现大气光晕效果;采用可见性判断和纹理精度管理的方法,实时判断地球表面区域的可见性、选择不同精度纹理粘贴、剔除不可见区域,减少绘制开销.实验显示,该方法在加载 1.02 GB 纹理像素的情况下,达到了 45 帧/s 的帧速率.

关 键 词: 虚拟地球;实时系统;虚拟现实

中图分类号: TP 391.9

文献标识码: A **文章编号:** 1001-5965(2011)08-1034-05

Realistic rendering of virtual earth in real-time application

Wang Cheng Dai Shuling

(School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: A new approach was proposed to enhance realism of virtual earth in VR-based real-time applications. A virtual earth which had 21 600 \times 10 800 pixels surface texture, bumping surface and shine of atmosphere was constructed. The whole virtual earth was cut into 128 parts and then the position of vertices, normal vectors, texture coordinate were calculated by geometry. Using the techniques of the mixture of the textures of earth, normal and clouds, it achieved relatively vivid surface of the earth. By moving and scaling the board which had atmosphere texture, it showed the shine of atmosphere. To achieve good real-time, it eliminated the unvisible parts by judging the parts' visibility in real-time, and chose different resolutions of texture. Experiments show that this approach is able to run 45 frames per second with 1.02GB texture pixels.

Key words: virtual earth; real time system; virtual reality

随着空间科学事业的快速发展,太空环境仿真成为虚拟现实技术和计算机图形学的研究热点,作为其中的重要内容之一,虚拟地球的研究与实现,很早就被提出并取得了很多成果,详见参考文献[1-7].

由于地球体积巨大,表面特征复杂,难以实时绘制高精度的地表、地貌,加上现有的一些实时仿真中往往侧重于系统结构与任务设计,对于真实感实现则简单模拟或忽略,尤其是地球表面的纹

理精度较低,影响了仿真系统的逼真性.在此背景下,本文提出一种具有高精度地表纹理显示的虚拟地球实现方法,旨在不影响系统实时性的情况下,提高视景仿真中虚拟地球的逼真性.

主要包括以下 4 个方面:①地表纹理精度的提高,如陆地、海洋、云层等纹理精度的提高;②地貌真实感的提高,如山川、海洋形成的地表轮廓凹凸感^[8]的实现;③随观察者位置变化的大气层明暗和光照效果;④纹理加载与显示的优化.

收稿日期: 2010-03-29

基金项目: 国家高技术研究发展计划资助项目(2009AA01Z333)

作者简介: 王 成(1983-),男,湖北石首人,博士生, www0041607@gmail.com.

1 真实感虚拟地球的创建

虚拟地球由 3 部分构成:①地球几何形状;②地表、云层纹理及云层 Normal 图的 3 层混合纹理;③大气层. 其创建流程如图 1 所示. 地球几何形状是基础部分,其它两层都是在其基础上进行添加和处理.

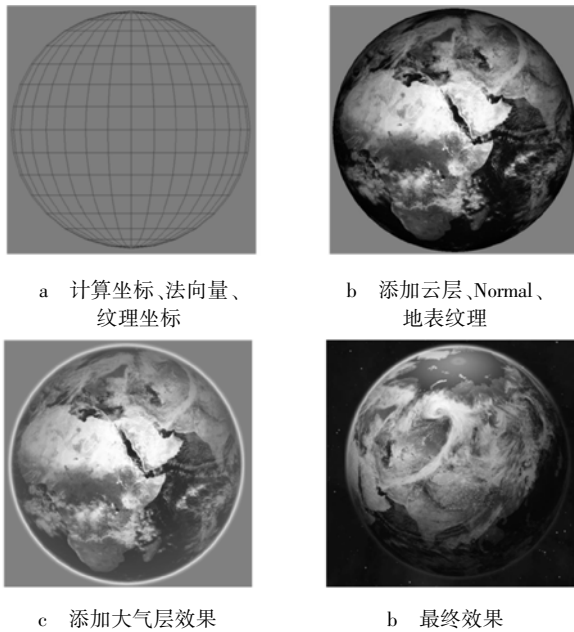


图 1 虚拟地球的创建流程

1.1 基于几何计算的地球形状模型构建

构建地球几何形状时,需要计算出几何球体表面若干个点的信息,包括顶点坐标、法向量、纹理贴图坐标等,供图形引擎^[9-11]绘制球体所用. 本文采用了顶点坐标和法向量计算与纹理贴图坐标计算分开的办法. 计算顶坐标和法向量时,将球体沿经线方向将球分为 16 块,每块跨越经度为 22.5°,沿纬线方向将球分为 8 块,每块跨越纬度为 22.5°,共将整个球体分为 128 块. 计算纹理贴图坐标时,将这 128 块中每一块又沿经纬方向各分为 6 块,即分为 36 小块. 顶点坐标、法向量、纹理贴图坐标的计算方法如下:

$$\begin{aligned}
 & i:(1 \sim 16); j:(1 \sim 8) \\
 & R = 6378164; \\
 & a = j * 22.5^\circ; \\
 & b = j * 22.5^\circ; \\
 & \text{positon}(i, j) = (R * \sin b * \cos a, \\
 & \quad R * \cos b, \\
 & \quad R * \sin b * \sin a); \\
 & \text{normal}(i, j) = \text{normalize}(p(i, j)); \\
 & i:(1 \sim 96); j:(1 \sim 48) \\
 & \text{texcord}(i, j) = ((i \% 6) / 6.0, (j \% 6) / 6.0).
 \end{aligned}$$

其中, R 为地球半径; $\text{positon}(i, j)$ 为球面顶点位置坐标; $\text{normal}(i, j)$ 为该顶点法向量; $\text{normalize}(p(i, j))$ 为将三维向量单位化; $\text{texcord}(i, j)$ 为顶点纹理贴图坐标.

这样处理的原因是:①为保证系统的实时性,球体划分的小块数量不能太多,因为块数越多,需计算和渲染的顶点就越多,势必会增加系统开销,而且在进行可见性判断时,判断的次数与顶点数成正比关系,也就是说,顶点数越多系统所需的实时计算也会越多,这样就增加了系统负担,影响仿真的实时性. ②为使球体圆滑,球体划分的小块数量不能太少,因为数量太少会使得球体看起来很粗糙,有棱角,并不圆滑,这样会影响仿真的逼真性. 为解决这个矛盾,考虑到图形显示中看到的最后效果是纹理贴图而不是几何的点线面,本文采用了顶点坐标和法向量计算与纹理贴图坐标计算分开的办法,计算顶点坐标和法向量时用分块较少,既能满足球体的几何构架需求,又能确保实时性和可见性判断时的合理计算量,计算纹理贴图坐标时对每小块再细化,使得每块球面区域的纹理贴图呈现圆滑的效果.

1.2 3 层混合纹理实现

构建地球的几何模型后,下一步需要对球体粘帖多层混合纹理,以提高其视觉真实感. 本文采用了纹理混合和 Normal Map^[10] 技术,将地表纹理、云层纹理及云层 Normal 图通过 Cg 代码对顶点着色器和像素着色器的控制计算,得到一个 3 层混合纹理,流程如图 2 所示.

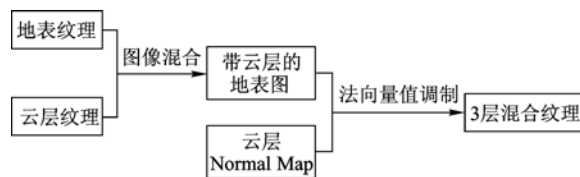


图 2 地表混合纹理实现流程

将地表纹理和云层纹理进行图像颜色的混合,得到带云层效果的地表图,读取云层 Normal 图的法向量信息,用信息值对该地表图进行颜色调制计算,得到了带 Normal Map^[10] 效果的地表图(如图 2 所示). 具体算法如下:

```

INPUT: EarthTex: 地表纹理图;
       CloudTex: 云层纹理图;
       CloudNormal: 云层 Normal 图;
       L: 光源位置;
OUTPUT: MixtureTex: 3 层混合纹理
{
    // 获取地表纹理图信息

```

```

float 4 einfo = tex2D ( EarthTex, texCoord.
uv );
// 获取云层纹理图信息
float 4 cinfo = tex2D ( CloudTex, texCoord.
uv );
// 获取 Normal 图的颜色通道信息
float 4 cinfo = tex2D ( CloudNormal, texCo-
ord. uv );
// 法向量调制和过滤优化处理
float 3 ninfo = saturate ( dot ( cinfo. xyz, L )
+ 0.3 ) * 1.1;
// 颜色融合
float 3 blend = 1 - ( 1 - einfo. rgb ) * ( 1 -
cinfo. rgb );
// 计算光照
float 3 diffuseColor = blend. rgb * ninfo;
// 输出颜色
MixtureTex = diffuseColor;
}

```

在实现时,将此算法放在图形硬件上来完成,充分发挥图形硬件的功效,提高计算效率.另外,为了实现逼真的视觉效果,在对片断着色器进行编程的过程中需要进行正确的法向量和光照计算,以获得正确的颜色信息.

相比单一的地表纹理图或者简单的地表纹理与云层纹理的混合图,3层混合纹理能够显示地球表面的轮廓凹凸效果,使仿真效果更加逼真.

1.3 大气层实现

本文中的大气层主要表现出光照情况下地表上空一段距离内大气对光的反射、散射、漫反射的模拟,阳照面和阴照面的明暗效果,以及大气半透明雾尘状的光晕效果.目前,虚拟现实一般采用粒子系统来实现大气层效果,但是众所周知,这种方法往往需要较多的预处理和调试工作,而且本文只关注从太空观察地球的场景,不需要从大气层观察太空,所以本文采用了一种简化的方法.采用实时移动与放缩带大气层效果的正方形面板来实现大气光晕效果,不仅可以取得不逊于粒子系统的效果,而且易于实现.

在一块正方形面板上(图3a)粘贴一张带有大气层效果的纹理图片(图3b),使视点与板中心连线始终与面板垂直,这样使得该图片始终朝向视线方向,即总是可见.将此板放到地球中心,按照大气层真实厚度调整尺寸以覆盖球体表面.当视点移动时,如图3c所示,面板Plane会在视点eye与地心O的连线L上移动,与球体相交出一

个圆环,根据圆的半径R对面板边长S按比例放缩,以确保面板与该圆的大小一致,使观察者能从其视角看到大气层厚度覆盖球体表面.虽然这样的大气层不具有物理真实性,但是从视觉真实感上来看,具有相当逼真的效果(图3d).正方形面板边长S计算方法如下:

$$L_1 = |\overline{OB}| \quad L_2 = |\overline{BC}| \quad L = |\overline{OC}|$$

$$L_1 = \sqrt{RR/L} \quad L_2 = L - L_1 \quad S = 2\sqrt{L_1L_2}$$

大气层的明暗效果是通过计算各顶点距离光源的远近确定亮度、计算各顶点法向量朝向确定该点是否背阴或朝阳,从而确定其明暗,再将该图片与球体表面纹理和星空背景进行颜色融合,产生半透明的雾尘状效果(图3d).

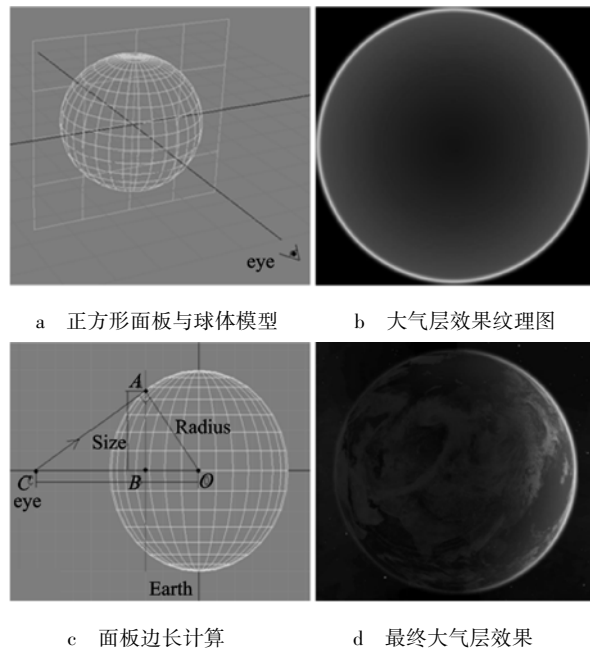


图3 大气层效果实现

2 高精度地表纹理的实时性能处理

由于地球尺寸巨大,为得到精细的画面,所粘贴的地表纹理也很精细,本文中实现时使用的地表纹理是128张1350像素×1350像素的图片,即总像素精度达到21600像素×10800像素,如此高的精度使之成为显卡加载图片信息时的瓶颈,导致系统实时性能低下.为优化计算,提高实时性能,本文采用了两种解决办法:①分块加载纹理,并采用实时的可见性判断,对可见部分加载显示,对不可见部分不加载;②采用纹理精度管理,根据场景真实度需要选择不同精度等级的纹理粘贴.

2.1 实时可见性判断

考虑到观察地球时,当视点处于距离地心较远时,比如2倍的地球半径处,最多只能看到半个

地球,当视点处于距离地心较近时,比如离地面 200 ~ 500 km 处,也只能看到 128 块中的 6 ~ 8 块,于是本文加入了可见性判断,实时计算并判断哪些区域是在视点可见范围内,对不可见部分进行剔除,这样虽然增加了计算开销,但避免了不必要的区域被渲染的开销,而且经测试这些渲染的开销是计算开销的近百倍,所以可见性判断很有必要,可以大大提高系统的实时性能.具体计算方法如下:

$$i:(1 \sim 16);j:(1 \sim 8)$$

$$v(i,j) = eye(i,j) - position(i,j);$$

$$value = dotproduct(v(i,j),n(i,j));$$

先计算当前视点坐标 $eye(i,j)$ 到每块区域顶点坐标 $position(i,j)$ 的向量 $v(i,j)$,再求得该向量与顶点法向量 $n(i,j)$ 的点积 $value$,其中,法向量 $n(i,j)$ 穿过球心和点 $position(i,j)$ 指向球外.若值为正,即两者夹角小于 90° 的即在视点可见范围内.求出可见顶点后,计算并判断出该顶点周围的 4 块区域,这些区域都为可见(如图 4 所示白色部分).

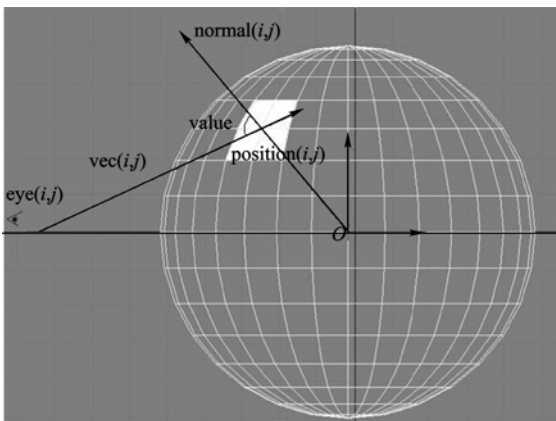


图 4 可见性判断

2.2 纹理精度管理

本文主要根据视点与观察对象的距离实时切换对象的相应精度级别的纹理.考虑到观察地球时,视点是变化的,球体上的 128 块区域与视点的距离也是变化的,只有当视点距离某块区域很近时,该块区域才有必要显示高精度的纹理,其它区域由于距离较远可以显示低精度级别的纹理,于是对每块区域设置不同精度级别的纹理,根据视点与该区域的距离远近,选择相应精度的纹理.这样减少了系统对加载纹理的开销,又不影响系统的逼真性.

本文纹理精度共分为 5 个等级,每次调整精度时并没有完全按照分辨率的 1/2 处理,而是根据实际需要进行适当的放缩处理(表 1).

表 1 纹理精度管理表

| 视点离地高度/km | MipMap ^[8] 级别 | 地表纹理精度/像素 | 云层纹理、云层 Normal 精度/像素 |
|---------------|--------------------------|---------------|----------------------|
| 0 ~ 200 | 1 | 1 350 × 1 350 | 750 × 750 |
| 200 ~ 1 000 | 2 | 675 × 675 | 375 × 375 |
| 1 000 ~ 2 000 | 3 | 450 × 450 | 250 × 250 |
| 2 000 ~ 7 000 | 4 | 270 × 270 | 150 × 150 |
| 7 000 ~ ∞ | 5 | 135 × 135 | 75 × 75 |

值得注意的是,由于单块区域纹理精度达到 1 350 像素 × 1 350 像素,加上同时显示多张,当视点快速移动运行时,频繁加载可见部分区域纹理和剔除不可见区域纹理,会有瞬时停滞.可以通过两种措施来改善:

- 1) 避免视点在多块区域的快速移动.本文中地球每块区域的经、纬跨度均为 22.5° ,在一些太空仿真相关的项目中,如:卫星在轨仿真,一般运行在离地面 200 ~ 500 km 的高空,要飞行跨越这一块区域需要几十分钟甚至几小时,所以,仿真中也不需要视点快速移动;
- 2) 做预处理,在初始化阶段将纹理信息从硬盘中读取到缓存中.

3 测试及结果分析

地球几何模型顶点数为 128 个,使用的地表纹理是 128 张 1 350 像素 × 1 350 像素的图片,总像素精度为 21 600 像素 × 10 800 像素 (667 MB),云层纹理、云层 Normal 均为 128 张 750 像素 × 750 像素的图片 (192 MB),合计纹理像素大小为 1.02 GB.

实现时硬件:PC 机 CPU 为 P4-2.8 GHz,显卡为 nVidia-Geforce6800,硬盘 120 G.运行平台为 WindowsXP.软件工具有:Visual Studio C/C++ 2005,OpenGL 的 glut 库,Cg,3dsmax9.0.

系统运行分为初始化与循环运行阶段,其流程图如图 5.

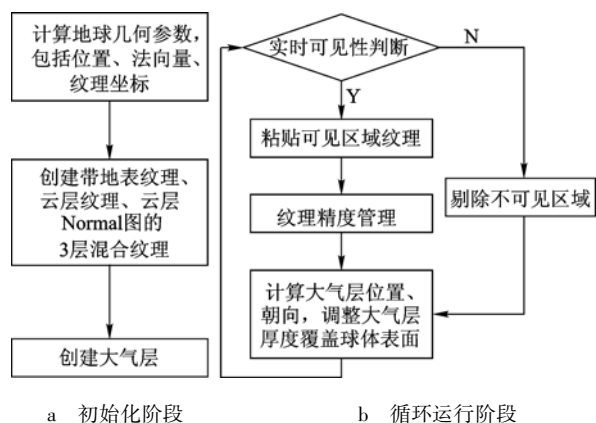


图 5 系统流程图

实时性能测试如表2所示.

表2 帧速率测试方案

| 视点离地高度/km | 可见区域个数 | MipMap 级别 | 帧速率/ (帧·s ⁻¹) |
|-----------|--------|-----------|------------------------------|
| 100 | ≤4 | 1 | 57 |
| 200 | ≤4 | 1 | 49 |
| 600 | ≤9 | 2 | 45 |
| 1000 | ≤16 | 2 | 47 |
| 2000 | ≤24 | 3 | 51 |
| 7000 | ≤42 | 4 | 62 |
| 10000 | ≤46 | 5 | 75 |
| 20000 | ≤60 | 5 | 78 |

表2中实验结果显示,系统帧速率始终保持在45帧/s以上,系统实时性良好.

虚拟地球的最终效果如图6~图8所示,分别是远瞰、俯瞰及全景效果.

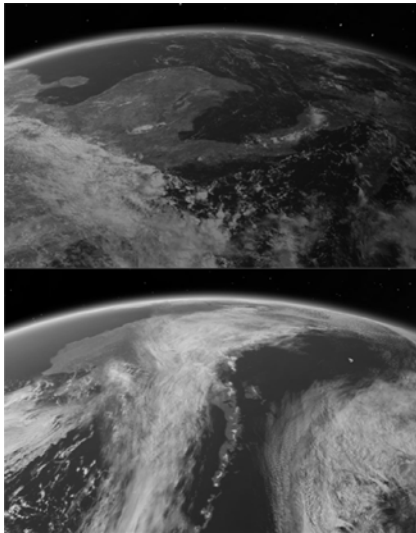


图6 远瞰效果图



图7 俯瞰效果图



图8 全景效果图

4 结 论

试验结果显示,本文所构建的虚拟地球取得了较逼真的效果,实现了高精度的地表纹理显示、地球表面轮廓凹凸效果和随观察者位置变化的大气层明暗和光照效果,实时性也很好,可用于与太空仿真相关的项目,作环境背景之用.

参考文献 (References)

- [1] 董加强. 基于 STK 的航天发射可视化仿真系统设计与实现 [J]. 计算机测量与控制, 2010, 18(2): 446-449
Dong Jiaqiang. Design and realization of aerospace launch visualization simulation system based on STK [J]. Computer Measurement & Control, 2010, 18(2): 446-449 (in Chinese)
- [2] Chen Aijun, Leptoukh G, Kempner S, et al. Visualization of a-train vertical profiles using google earth [J]. Computers & Geosciences, 2009, 35(1): 419-427
- [3] Thorne C, Weiley V. Earth's avatar: the web augmented virtual earth [C] // Session: Visualisation. SIGGRAPH 2003 Conference on Web Graphics. San Diego: ACM, 2003: 1-11
- [4] Gray J, Szalay A. The world-wide telescope [J]. Communications of the ACM, 2002, 45(11): 50-55
- [5] 沈荣骏. 数字地球, 在应用中求发展 [J]. 装备指挥技术学院学报, 2009, 20(11): 1-6
Shen Rongjun. Digital earth, making progress through applications [J]. Journal of the Academy of Equipment Command & Technology, 2009, 20(11): 1-6 (in Chinese)
- [6] 曾鹏, 陈长征, 李苏军. 基于数字地球的虚拟海战场环境仿真 [J]. 计算机工程, 2009, 35(8): 269-273
Zeng Peng, Chen Changzheng, Li Sujun. Simulation of virtual sea battlefield environment based on digital earth [J]. Computer Engineering, 2009, 35(8): 269-273 (in Chinese)
- [7] 李亚臣, 蒋红柳, 熊海林, 等. 视景仿真中三维地球的建模 [J]. 计算机工程, 2007, 33(12): 225-227
Li Yachen, Jiang Hongliu, Xiong Hailin, et al. 3D earth modeling in visual simulation [J]. Computer Engineering, 2007, 33(12): 225-227 (in Chinese)
- [8] 谢丰, 姜毅. Mipmap 应用于纹理绘制的新算法及实现 [J]. 应用科技, 2008, 35(4): 57-60
Xie Feng, Jiang Tao. New algorithm and realization of mipmap for texture rendering [J]. Applied Science and Technology, 2008, 35(4): 57-60 (in Chinese)
- [9] 舒畅, 梁英伦, 王大为. 基于 3Dmax 和 OGRE 的某型装备视景仿真研究 [J]. 指挥控制与仿真, 2009, 31(5): 78-80
Shu Chang, Liang Yinglun, Wang Dawei. Research on an equipment visual emulator based on 3Dmax and OGRE [J]. Command Control & Simulation, 2009, 31(5): 78-80 (in Chinese)
- [10] Michael Toksvig. Mipmapping normal maps [J]. Journal of Graphics, GPU & Game Tools, 2005, 10(3): 65-71
- [11] Hu W, Qin K. Interactive approximate rendering of reflections, refractions and caustics [J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(1): 46-57

(编辑: 刘登敏)