

基于图像的异龄树木模拟研究*

杨会君^{1,2}, 周晓峰¹, 何东健¹, 王昕³

(1.西北农林科技大学 机械与电子工程学院, 陕西 杨凌 712100; 2.西北农林科技大学 信息工程学院, 陕西 杨凌 712100; 3.西北农林科技大学 外语系, 陕西 杨凌 712100)

摘要:提出了一种交互式变换树种和地面的方案,并结合微软公司新推出的Direct3D技术和VC++开发环境,实现了不同种树木在不同地表生长环境中的真实性模拟,建立了虚拟异龄树木生长环境模型和异龄多种模型,避免了传统的几何建模对时间和空间的耗费,帮助林业从业人员在普通的微机上对森林进行有效的监测和管理,从而提高林木生产量和资源利用率。

关键词:世界变换; 视角变换; 投影变换; 场景渲染

中图分类号:TP391.41

文献标识码:A

Simulation research of uneven-aged trees based on image

YANG Hui Jun^{1,2}, ZHOU Xiao Feng¹, HE Dong Jian¹, WANG Xin³

1. College of Mechanical and Electronic Engineering, Northwest A&F University, YangLing 712100, China;

2. College of Information Engineering, Northwest A&F University, YanLing 712100, China;

3. Department Foreign Languages, Northwest of A&F University, Yangling, 712100, China)

Abstract: The thesis combines corresponding picture process technology, researches the reconstruction of trees, three dimensional model from the two-dimensional picture, proposes one solution that can interactively transform the species of tree or the surface, finally adopts the three-dimensional animation technological Direct3D with VC development environment to realize the authentic simulation of the different species of trees in different surface habitat. Render a virtual environment where different age of trees grow and model all kinds of trees, that users can change the species of trees and surface, and offer forestry workers parameters of trees. This paper avoids exhaust on space and time that traditional geometry-based modeling method has arose. So make forestry workers carry on effective monitoring and management to the forestry and improve the amount of production of forest and resource utilization ratio.

Key words: world transformation; view transformation; projection transformation; scene rendering

目前,我国森林存在面积少、草场退化、水土流失和土地荒漠化等问题,对林木生长环境的模拟限于人工试验阶段,周期较长且浪费人力、财力,给林业研究者带来极大的困难和不便。解决这些问题的关键在于深入了解森林的生长发育规律。目前我国主要采用3S技术、数字通信、机械自动化、传感器技术和林木遗传工程等建立森林的生态模拟环境^[1-2],这些建模方法主要

缺点是集中在单株树木模拟^[3-4],其建模、造型显得过于复杂^[5],且当前的模拟技术带有主观的人为因素^[6-8],影响林木自然生长状态的模拟。因此,开发本系统以实现森林生长环境的真实性模拟,帮助林业工作者对森林资源进行有效的管理与监测,改变过去以步行实地考察的造林模式,有利于挖掘和发挥林地的生产潜力,最大限度地发挥林地的生产效率;有利于缩短森林的生

*基金项目:西北农林科技大学人才基金(01140501);农业推广项目

长周期，加快森林资源的培育速度，改善我国的生态环境。

本文研究异龄多种树木生长环境的真实性模拟。主要运用 Direct3D 的三维立体模式结合 VC++ 的开发环境进行基于图像处理的应用研究^[9]，实现异龄多种树木生长环境的空间非空间模拟场景的实现。在场景的真实性模拟过程中，首先研究场景绘制过程中的变换流程，把采集到的图像处理成为形象逼真的天空纹理、地表纹理、树种纹理；在场景中变换树种的同时变换地表，从而观察不同生长环境中树木的生长状况；应用基于图像实现图形绘制的技术，即 dds、bmp 和 x 等纹理材质技术，以帮助林业工作者及时掌握林业采伐的生产进度、计划进度、伐区拨交数、出材量和作业质量。

1 算法设计与实现

1.1 方案选择

基于图形的技术很容易实现异龄树木的空间漫游，但树种和地表的变换都不易实现。因此，本文选择基于图像的技术，利用 VC++ 的开发环境结合 D3D 中对纹理文件(mesh 和 dds 文件)的处理机制进行图像处理和分折。DirectX 9 中的 PS 2.0 可以支持 160 个硬件指令，同时操作 16 个材质数量，能对纹理效果实时演算、动态纹理贴图，不占显存，理论上对材质贴图的分辨率精度无限高，新的高精度浮点数据规格可以使用多重纹理贴图，可操作的指令数可以任意长^[9]。

保留模式中使用执行缓冲，允许在使用执行缓冲的同时使用留模式的 API 函数。本系统通过把执行缓冲看作是 D3D 中的可视对象来处理。首先，创建和管理标准 Windows 窗口应用程序以执行基本保留模式初始化，在系统主文件处理消息循环和物体图形显示，形成了 D3D 保留模式应用程序的一般框架。包括创建 D3D 对象，创建主要的场景和摄像机，改变渲染品质等。在 D3D 保留模式中用循环渲染的方法把对象渲染到屏幕上。首先，对所有框架应用旋转和速度，然后调用清除当前的视口并设置背景颜色。之后把当前场景渲染到当前视口上，以复制渲染的图像来显示。

1.2 系统整体流程设计

在系统设计中，首先要对采集到的树种和地表图像进行处理，用 α -通道技术将树种图像转换为 dds 格式的纹理文件，地表图像用 PandaDXExport4.6.62.0.dll 或 conv3ds 转换为 X 格式的纹理文件，再用 D3D 技术进行场景的设置。场景实现模块主要包括纹理文件的渲染及场景变换的实现，具体流程如图 1 所示。

在界面设计中，系统首先对场景进行初始化，以实现控件和纹理的初始化，然后用图像建模方法构造树

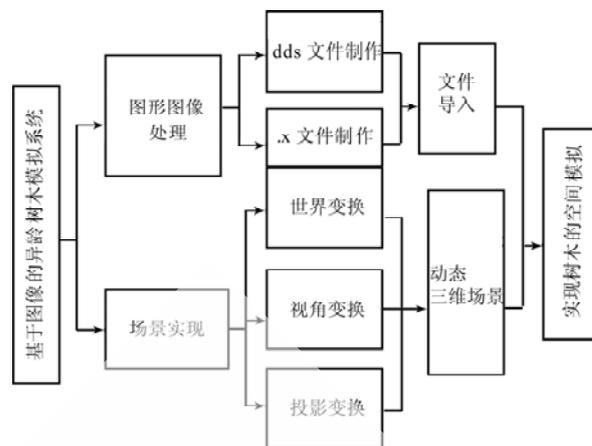


图 1 系统模块和结构设计图

种模型和纹理，并对其进行渲染，最后实现场景变换。

1.3 各模块实现方法

1.3.1 树木模型建立方法

本设计中采用结构体定义树木的属性模型。为了方便纹理贴图，需要创建一个矩形缓冲区存放渲染树木所需要的图像模型同时制作 dds 和 bmp 格式文件纹理材质。dds 文件在制作时应注意前景色与背景色必须分明，以利于 α -通道技术将图像处理为 dds 文件。为实现树种在场景中的变换，确保初始化的形式与重新赋值的形式一致，用字符串拷贝方式对 dds 进行初始化。最后将纹理与图像模型融合实现场景中显示。

1.3.2 纹理材质的绘制与处理方法

本设计将纹理处理为 COM 对象，纹理的绘制要经过创建纹理、载入内存设置纹理、设置纹理渲染状态(纹理混合)3 个过程。可从现有的图像中直接创建，也可以创建一个空的纹理对象，然后再进行填充。为了计算方便，图片的大小处理为 2n 大小，设置纹理的过程中采用的纹理过滤算法有：树木纹理采用最近点采样过滤和线性纹理过滤；地表纹理采用各向异性纹理过滤和多级渐进纹理过滤^[10]。在实现多种材质变换时，材质接口指针首先释放原来的材质缓存。根据纹理文件名称创建纹理指针，开辟新的缓冲区存储纹理，并将纹理指针设置为当前纹理。最后在场景渲染函数中使用，即可实现多材质的变换。

在 D3D 中一个包含了顶点信息、材质、贴图、动画帧的 3D 对象称为 mesh 物体，将该对象的信息提取存放在文本文件中，称为 x 文件。x 文件制作时，首先将渲染的物体的顶点、贴图、材质信息提取，然后把它的坐标(3D 坐标)转换为屏幕坐标，再添加一些灯光信息。最后调用渲染方法，根据各个顶点的位置(屏幕坐标)，用灯光和材质信息计算亮度，把贴图“填”进去，这样得到的显示效果立体感强^[11]。x 文件制作成功后，便可在

D3D 中通过类或接口指针进行创建调用。

1.3.3 场景变换实现方法

在本系统中, 场景中的所有对象通过下面 3 种形式的变换, 实现最终显示: (1)世界变换(World Transformation): 在建立三维实体的数学模型时, 通常以实体的某一点为坐标原点构成坐标系, 称为本地坐标系。所研究的实体总是位于某个场景中, 而场景采用世界坐标系, 因此需要把实体的本地坐标变换成世界坐标, 称为世界变换。本文通过一个 4×4 矩阵来实现, 对于世界变换, 只需知道实体在场景中的位置信息, 就可以得到变换矩阵。多次变换用矩阵相乘的方法, 逆变换用逆矩阵实现。(2)视角变换(View Transformation): 实体位置确定后, 接下来要确定观察者在世界坐标系中的方位, 即在世界坐标系中如何放置摄像机。确定观察者需要 3 个量: 观察点的坐标、视线方向(本文用视线上的一个点来替代, 此时视线方向就是从观察点指向该目标点)、上方向(观察者的正上方向)确定后, 以观察者为原点, 视线为 Z 轴, 上方向为 Y 轴, 构成了视角坐标系, 把实体从世界空间转换到视角空间, 这个坐标变换称为视角变换^[6]。(3)投影变换(Projection Transformation)实体转换到视角空间后, 还要经过投影变换, 三维的实体才能显示在二维屏幕上。本文使用透视投影变换, 此时在视角空间中, 可视区域是 1 个以视线为轴心的棱台, 如图 2 所示。想象用户处在 1 个没有可见度的房间里, 面前有 1 扇窗户, 你可以透过窗户看到各种景物。窗户就是棱台的前裁剪平面, 天空、远山等背景是后裁剪平面。投影变换把位于可视棱台内的景物投影到前裁剪平面, 由于采用透视投影, 距离观察者远的对象会变小, 从而更具真实感。在系统中, 前裁剪平面被映射到程序窗口, 最终形成了在屏幕上看到的画面^[11]。透视变换由 4 个量决定: 前裁剪平面的宽度 w ; 前裁剪平面的高度 h ; 前裁剪平面到原点的距离 z_1 ; 后裁剪平面到原点的距离 z_2 ^[12]。

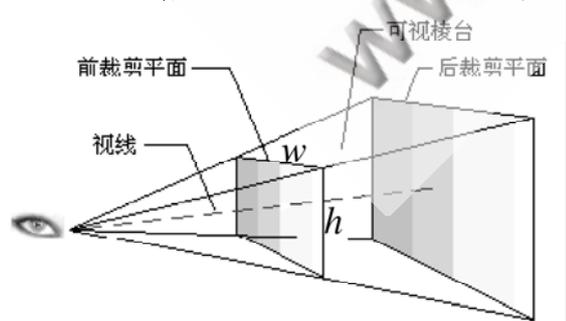


图 2 场景中可视区域

1.4 场景绘制主流程

在 D3D 中, 本文提供了一套完整的流程用来设置场景、操作 3D 物体、渲染画面。实现流程的主要函数有:

(1) OneTimeSceneInit(), 该函数可以对场景中需要的材质或纹理进行初始化, 也可对场景的可用空间进行初始化; (2) RestoreDeviceObjects(), 存储设备函数, 用于在设备重新设置后恢复设备对象; (3) FrameMove(), 用来转换视角, 实现世界变换后的场景更新, 设置投影变幻参数; (4) Render(), 渲染函数, 用来绘制天空、地面场景, 渲染树木对象, 实现投影变换; (5) Delete- DeviceObjects(), 删除 RestoreDeviceObjects() 函数中恢复的设备; (6) FinalCleanup(), 释放 OneTimeSceneInit() 函数初始化的接口、指针。

1.4.1 渲染和清空表面

在场景绘制的过程中, 物体进行纹理渲染对实现三维逼真效果非常重要。在渲染过程中, 首先用 BeginScene 方法通知 D3D 场景的渲染将要开始, 使系统检查它内部的数据结构, 渲染表面的有效性和正确性, 并设置内部标志来标示一个场景正在渲染中, 然后调用不同的函数来渲染场景中的图元或组成对象的顶点。最后, 渲染完一个场景之后, 用 EndScene 方法结束场景渲染, 清除标示场景正在进行的内部标志, 清空 cache 数据, 并检查渲染表面是否完整。所有的场景渲染方法都必须在 BeginScene 和 EndScene 之间调用。以防表面丢失或出现内部错误。本文不允许在场景中嵌入另一个场景, 即必须渲染完一个场景之后, 再开始渲染另一个场景。当一个场景正在渲染时, 不能使用 GDI 操作, 否则, 会使 GDI 操作的结果不可见。如果程序要使用 GDI 函数, 一定要确保所有 GDI 调用在场景函数之外。场景渲染部分关键代码如下:

```
if(SUCCEEDED(lpD3D->BeginScene( )))
{ // Render primitives only if the scene started successfully.}
// Always close the scene, even if BeginScene fails.
hr = lpD3D->EndScene();
```

在渲染场景中的对象之前, 必须清空渲染目标表面(render target surface)上的视口或视口的子集。渲染目标表面的清空操作能够为需要的区域设置一种缺省的颜色或纹理, 对于深度和模板缓冲, 能够设置一个深度或模板值^[12]。本设计清空视口时, 首先设置渲染目标表面的所需部分, 主要是相关的深度或模板缓冲状态, 使渲染的表面区域重新复位。

1.4.2 多树种和地表变换

多种树木和地表的变换实质上就是多材质贴图的实现过程。首先实现多材质文件从磁盘上的导入、初始化和加载。在本文中, 创建 DrawTree() 函数作为多树种变换的感应函数, 在多材质贴图变换过程中, 当在材质缓冲区内接收新的材质时, 必须使用安全释放函数释放缓冲区里的旧材质。

2 结果与分析

本文研究基于图像建立异龄树木模型并绘制其生长空间的虚拟环境,实现了预期的功能。不仅能展现不同的树种的生长状况,而且实现树木在不同环境中生长,并在每种生长环境中,都能观察不同年龄阶段的树木的生长过程。

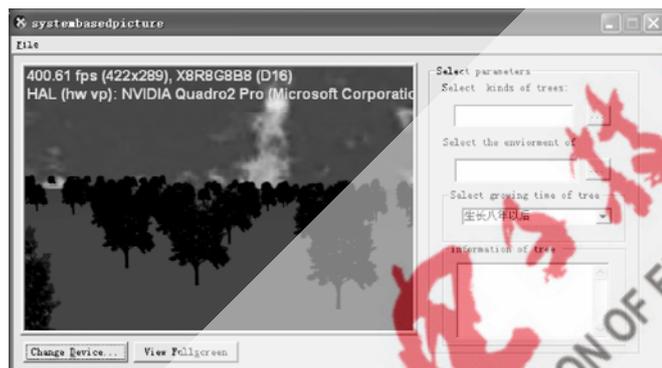
2.1 系统测试的软硬件环境

硬件环境: CPU 主频 1.7 GHz, 内存 DDR 512 MB, 硬盘 40 GB, 显卡: 刷新频率 75 Hz, 色彩位数 24; 显示分辨率 1 024 × 768。

软件环境: 操作系统 Windows 2000 或 Windows XP、Direct3D、VC++。

2.2 测试效果及分析

树种变换: 本文在异龄树木模拟环境中,可以由用户自己选择需要模拟的不同树种,从而在相同生长环境中实现不同树种之间的变换,如图 3 所示。



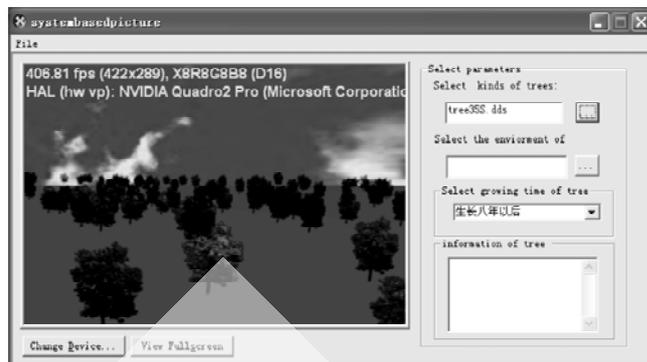
(a) 变换树种之前



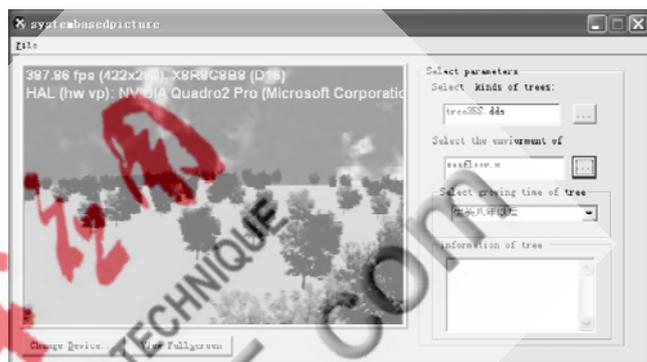
(b) 变换树种之后

图 3 不同树种生长空间模拟

地表变换: 基于实际的需求,为体现树木在不同环境中生长状况的差异,系统实现了单种树木在不同环境中的生长状况,用户可以根据需要选择合适的生长环境,如图 4 所示。



(a) 绿荫地环境中生长



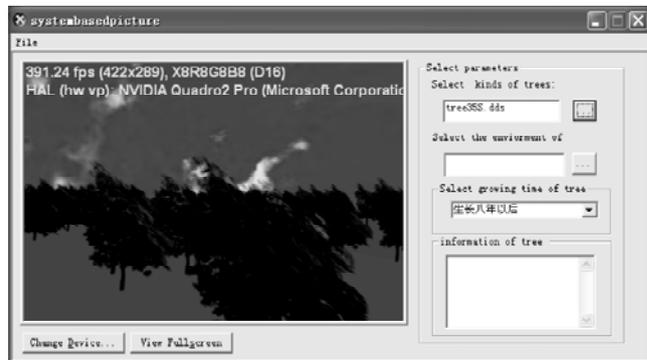
(b) 沙地环境中生长

图 4 树木在不同环境生长模拟

异龄树木生长状况: 在虚拟环境中,用户可以选择不同年龄阶段,从而便于研究和观察树木在不同生长阶段的生长状况以及对空间的占用情况,如图 5 所示。



(a) 4年后生长状况



(b) 8年之后生长状况

图 5 异龄树木生长状况模拟

(下转第 53 页)

(上接第 49 页)

为使系统尽可能达到优化, 本文所设计的系统具有系统性、实用性、通用性以及可扩充性, 从而便于系统的改进和扩充, 使系统处于不断完善的过程中。

本文从基于图像的角度研究了异龄树木的空间模拟系统的开发方法、模块设计、细节纹理的处理并最终实现系统的全过程。帮助林业工作者在普通的微机上实现对异种树木生长环境进行研究, 它可以真实地模拟异龄树木对生长环境的空间占用现状, 弥补了我国林业工作者长期以来在林木生长环境研究中, 手工操作的低效率和低准确性等不足之处, 使林业领域的研究更加科学、更加准确。同时为管理者节约大量购买专用图形工作站的成本。系统具有再现性、容易控制、处理多样性、交互性等特点。但是, 系统在树木的定位观测和用户交互性方面还存在了一些不足, 还需要进一步改进。

参考文献

- [1] 权兵. 基于虚拟森林环境的林分生长和经营模拟研究[D]. 福州大学, 2005.
- [2] 宋成芳, 谈奇峰, 张龙, 等. 风场作用下的动态森林场景的实时仿真[J]. 计算机辅助设计与图形学学报, 2007, 119(13): 323-328.

《电子技术应用》 www.ChinaAET.com

- [3] 雷相东, 常敏, 陆先昌. 虚拟树木生长建模及可视化研究综述[J]. 林业科学, 2006, 142(111): 123-131.
- [4] 谢绍锋, 肖化顺. 虚拟森林资源信息三维可视化中树模型的生成方法[J]. 中南林业调查规划, 2007, 26(1): 43-47.
- [5] 舒娱琴, 李卫红. 面向虚拟森林经营管理的树木交互式参数建模方法[J]. 林业科学研究, 2007, 20(3): 338-343.
- [6] DAVID J.K. Programming Visual C++ (5th) [M]. Microsoft Press: 2008.4.
- [7] 唐丽玉, 陈崇成, 权兵. 森林景观的计算机建模与可视化研究进展[J]. 林业科学, 2006, 42(10): 109-116.
- [8] 河南平, 周翠竹. 基于 ArcObjects 的树木景观建模[J]. 测绘与空间地理信息, 2006, 29(1): 33-35.
- [9] 王玉增, 王秀明, 盖旭升. 基于 DirectX 的虚拟驾驶模拟视景系统设计方法[J]. 系统仿真技术, 2008, 4(1): 1-5.
- [10] DirectX document[M]. Microsoft. 2007.
- [11] X 文件的使用. <http://data.gameres.com/document.asp?TopicID=47493>[EB/OL]2006.
- [12] ANDRE L. Tricks of the 3D game programming gurus[M]. Sams. 2005.

(收稿日期: 2009-03-17)

《信息化纵横》2009 年第 12 期