基于分类及环境特征的树木真实感绘制*

陈彦云[1], 严 涛[1], 张晓鹏[2], 吴恩华[1,3]

[1](中国科学院 软件研究所 计算机科学开放研究实验室,北京 100080);

[2](中国科学院 自动化研究所,北京 100080);

[3](澳门大学科技学院,澳门)

E-mail: cyy@ox.ios.ac.cn

摘要: 生成高度真实感的虚拟自然场景一直是图形学研究领域中的一个富有挑战性的难题.作为自然场景的重要组成部分,树木的真实感模拟也得到人们的广泛重视.树木种类繁多,形态各异,复杂的结构使其无论在造形、存储还是在绘制上都存在着相当的困难.针对不同的环境特征和不同类型的树木,根据它们的具体特点需采用不同的绘制手段.对于阔叶树,采用基于OpenGL的深度缓存阴影生成算法;对于针叶树,则采用结合光线跟踪和纹元(texel)绘制的技术;而对于距离视点比较远的树,采用的是体纹理映射(volumetric texture mapping)的方法.实践证明,这几种绘制技术基本上可以满足各种不同类型场景的树木的绘制要求.

关键词: 真实感图形生成;自然场景;纹元;体纹理

中图法分类号: TP391 **文献标识码**: A

树木等自然景物的模拟,一直是计算机图形学领域中具有挑战性的研究方向之一.自20世纪60年代起,人们从不同的角度来构建模型以模拟树木的形态,使得这方面的研究有了长足的发展.1975年,A. Lindenmayer提出了所谓的L系统[1],并与P. Prusinkiewicz合作将其用于树木和其他植物的造型[2];Aono和Kunii引入枝间的相互吸引、抑制作用及其夹角的统计变化机制,对子干在母干上的分布做了详细的估算[3];De Reffye等人提出严格的植物学模型[4];Holton还考虑到重力的作用[5];Oppenheimer运用分形技术对树木进行造型[6];Reeves等人用粒子系统方法模拟作为背景环境的草地、森林[7];Weber等人从整体几何结构出发,用若干参数对树木造型进行控制,并对森林的绘制进行了探讨[8].以上方法都取得了良好的效果,树木造型的理论也随之日益成熟起来.

随着造型手段的不断丰富和完善,有效地绘制树木越来越得到人们的重视.到目前为止,树的真实感绘制手段可以大致分为以下3种:

(1) 绘制用粒子系统构造的树;

(2) 直接绘制多边形等几何面所构造的树;

(3) 绘制由二维纹理构造的远距离的树.

由Reeves等人提出的粒子系统所绘制的森林是早期计算机生成虚拟自然景物的最有说服力的例子之一,然而粒子系统构造的树木看起来有比较明显的人造痕迹,而且粒子系统的光照效果无法根据客观事实严格计算[7],因此,缺乏高度真实感.

收稿日期: 1999-08-09; **修改日期**: 1999-10-20

基金项目: 国家自然科学基金资助项目(69873044);Research Grant of University of Macao (RG 009/99-005/WEH/FST)

作者简介: 陈彦云(1971-),男,广东潮州人,博士生,主要研究领域为真实感图形,计算机动画;严涛(1971-),男,辽宁大连人,博士生,主要研究领域为计算机图形学;张晓鹏(1963-),男,陕西西安人,博士后,主要研究领域为计算图形学,CAD,医学图像;吴恩华(1947-),男,江苏南通人,研究员,博士生导师,主要研究领域为计算机图形学、虚拟现实,科学计算可视化,计算机动画.

对由多边形等几何面所构造的树的绘制可以采用成熟的真实感图形绘制手段[9,10],如各种光照模型、z-buffer 深度消隐和阴影生成方法、光线跟踪方法、辐射度方法等.考虑到场景的复杂性,一般采用 z-buffer 深度消隐和阴影生成方法来绘制近距离的阔叶树.对于几何面所构造的远距离的树,Weber 采用的是对模型进行简化,根据树木的远近采用多分辨率的绘制手段[8].虽然 Weber 提供的绘制结果质量比较高,但是远处树木简化后颜色过于单一,造成绘制结果模糊不清[11].

三维纹理是最近几年才被引入到树木的构造和绘制中的[11~13].它的优点是可以简化场景,消除几何面造成的混淆,使构造和准确绘制高度复杂的自然场景成为可能.它的缺点则是由于绘制算法通常是结合体绘制技术的光线跟踪方法,因此实现起来比较复杂,运算时间复杂度也比较大.

自然界的树木种类繁多,形态各异,考虑到存储容量和绘制速度的限制,针对不同的环境特征和不同类型的树木的特点,我们采用不同的绘制手段:对于只包含近距离阔叶树的简单场景,我们采用基于 OpenGL 的深度缓存阴影生成算法,实现简单,绘制速度快;对于近距离针叶树,我们采用纹元(texel)来构造松针束,并在绘制时采用结合光线跟踪和纹元绘制的技术;而对于距离视点比较远的树,为了简化场景,我们采用的是体纹理映射的技术,因此绘制方法采用的是结合体绘制技术的光线跟踪方法.从各种绘制方法所得到的结果图像不难看出,根据不同的场景要求,采用不同的绘制技术绘制树木是有效、可行的.

1 阔叶树简单场景的绘制

阔叶树简单场景是指场景中只包含一棵或者几棵阔叶树,这样的场景完全可以用多边形来描述,因而绘制时可以采用简单的深度缓存阴影生成算法.

1.1 深度缓存阴影生成算法的基本原理

深度缓存阴影生成算法分两步完成:首先,以光源为视点,将场景中的所有多边形对光源作带深度消隐的投影,并将结果的深度值记录在阴影深度缓冲区中;然后,恢复视点,采用带深度消隐的局部光照模型绘制场景,在绘制每个像素时,先将其投影到光源上,并与阴影深度缓冲区中对应的点作深度比较,如果当前绘制的像素离光源近,那么说明它落在阴影外面,否则就说明落在阴影中[14~16].

这个算法实现简单,处理速度快,但缺点是离散化可造成严重的混淆,尤其是在绘制准确定义的几何形体时这种混淆更明显[14].另外,对于场景中的每一个光源,都必须做一次投影操作,并记录一个阴影深度缓冲区,内存开销比较大.对于自然场景而言,这两方面的缺陷都可以接受,一是自然景物(包括树木)的结构高度随机,大大减小了混淆的影响;二是通常只包含一个光源——太阳,所以只需做一次对光源的平行投影操作,记录一个阴影深度缓冲区即可.

1.2 基于 OpenGL 的深度缓存阴影生成算法

采用 OpenGL 的原因是,它是一个广泛应用的图形基本函数标准,可以得到硬件的支持以提高绘制效率.我们提出的基于 OpenGL 的深度缓存阴影生成算法,就是要在涉及投影和光照计算以及深度消隐等操作上尽可能地采用基本 OpenGL 函数完成.根据上述深度缓存阴影生成算法的基本原理,相应地,基于 OpenGL 的算法可以通过以下几个步骤来完成:

(1) 将视点定在光源处,将场景中所有的多边形向光源作带深度消隐的平行投影,每个多边形采用一种颜色填充,这种颜色我们称为 IDcolor.将投影结果和深度信息保存在 SZ-buffer 中;

(2) 恢复视点,采用相同的 IDcolor 将场景中所有的多边形向视点作带深度消隐的透视投影,将屏幕上每个像素点的 IDcolor 和世界坐标记录在 SC-buffer 中;

(3) 运用基本 OpenGL 函数对场景进行局部光照绘制,得到结果图 A;

(4) 只保留环境光,运用基本 OpenGL 函数对场景进行局部光照绘制,得到结果图 B;

(5) 判断图 A 上的每个像素点,如果落在阴影中,就把图 B 中的对应点颜色值赋给它.

前 4 步都很容易实现,关键是第 5 步.因为图 A、图 B 与 SC-buffer 所保存的 IDcolor 及三维世界坐标一一对应,这样,在检查某个像素点是否落在阴影中时,假设在 SC-buffer 中该点的三维世界坐标为 $p(x,y,z)$,颜色为 IDcolor1,将 $p(x,y,z)$向光源作平行投影,获得的深度为 depth1,投影结果在 SZ-buffer 中所对应的 IDcolor 为 IDcolor2,深度为 depth2,那么如果满足下面 3 个条件之一,就表明该点落在阴影之外:(1) depth1<depth2;(2) IDcolor1=IDcolor2;(3) 平行投影结果超出 SZ-buffer.

对于局部光照绘制结果图中落在阴影中的像素点,把它的颜色值置为图 B 中对应点的值,否则置为图 A 中对应点的颜色值,因此,最后的结果图是由图 A 和图 B 合成的.

1.3 深度缓存阴影生成算法的绘制结果

图 1~5 分别为深度缓存阴影生成算法的 5 个步骤所对应的运算结果.

Fig. 1 ID color in SZ-buffer

图 1 SZ-buffer 的 ID color 图

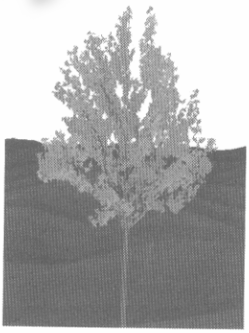

Fig. 2 ID color in SC-buffer

图 2 SC-buffer 的 ID color 图

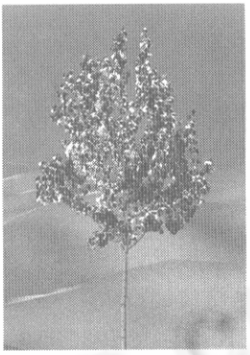

Fig. 3 Rendering result A, using primary OpenGL functions

图 3 OpenGL 基本函数绘制结果图 A

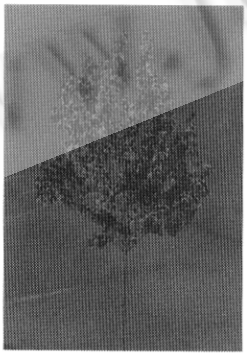

Fig. 4 Result B, using ambient only

图 4 只保留环境光的结果图 B

Fig. 5 Final result

图 5 最终结果

这种算法实现简单,绘制速度快.当我们采用这种算法进行树在风中摇曳的动画绘制时,在400 Hz Pentium II 的PC机上,对一百万个带纹理映射的三角形构成的场景的绘制其速度大约为30s/帧,绘制效果比较理想[17,18].下面的图6和图7是用这个算法绘制的另外两个比较复杂的场景的结果.

Fig. 6 Poplars in sunset
图6 晚霞中的杨树

Fig. 7 Willow in sunshine
图7 阳光下的柳树

2 针叶树简单场景的绘制

针叶树简单场景是指场景中只包含一棵或几棵近距离的针叶树.这时,深度缓存阴影生成算法不再适用,因为针叶树的叶子非常细小,而且数目巨大,采用深度缓存阴影生成算法将造成无法接受的混淆误差.因此,针叶树不能像阔叶树那样用多边形来描述.对于这类场景,我们采用的造型和绘制技术,是将纹元(texel)与光线跟踪二者结合起来的方法.

纹元是Kajiya在1989年提出的构造和绘制短皮毛的有效工具[19].因为松针细长,而且成束存在,与皮毛类似,所以在对原始纹元结构作调整后,可以用来描述松针束.

2.1 纹元的概念

最初由Kajiya和Kay提出的纹元,是指用来近似表示许多微面总体光学特性的一个三维参数数组[19].纹元数组空间中的任意一个体元(因为纹元数组与数据场类似,我们将纹元数组的单元也称为体元)都存储着如下3个部分的内容:(1) 密度标量 ρ,表示该体元包含的所有微面的近似投影面积;(2) 结构场 B,表示该体元中各微面的局部方向;(3) 光照模型场 Ψ,它决定着将有多少光线从微面散射出来.换句话说,纹元包括(ρ,B,Ψ)这3个部分,是散射密度 $\rho(x,y,z)$、结构组合 $B=[n(x,y,z),t(x,y,z),b(x,y,z)]$以及双向光线反射场的函数:

$$\Psi(x,y,z,\theta,\varphi,\psi).$$

散射密度 ρ 表示体元有多大的投影面积被微面覆盖.结构场组合 B 是一些基本的坐标向量,分别是法向量场 n、切向量场 t 和副法向量场 b.双向光线反射场函数 Ψ 表明体元包含的是何种曲面.

作为三维数据场,纹元在绘制时采用的是体绘制技术[13,19].根据Kajiya和Kay在文献[19]中的描述,纹元的光照方程为

$$T=e^{-r\sum_{S=P_{near}}^{P_{far}}\rho(x(s),y(s),z(s))}, \quad (1)$$

$$B=\sum_{d=P_{near}}^{P_{far}}\left\{e^{-r\sum_{u=P_{near}}^{d}\rho(x(u),y(u),z(u))}\times\sum_{i}[I_i(x(d),y(d),z(d)\Psi(x(d),y(d),z(d),\theta,\varphi,\psi)]\times\rho(x(d),y(d),z(d))\right\}. \quad (2)$$

式中 I_i 是光源 i 的强度;P_{near} 和 P_{far} 是视线与纹元包围盒的两个交点,如图 8 所示.

尽管上面介绍的纹元的理论模型很复杂,但是在实际应用中,这个模型是可以简化的.比如在 Kajiya 绘制玩具熊时,由于表示玩具熊毛发的数据场中只有相互平行的毛发段,因此结构组合 B 和反射函数 Ψ 显然都是恒定的.所以,只有表示散射密度 ρ 的非透明度才是真正需要存储的量[13].

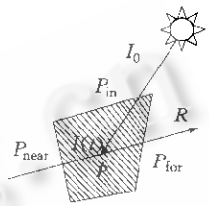

Fig. 8 Ray-Texel intersecton
图8 光线和纹元相交

我们采用与 Kajiya 的纹元类似的结构来描述松针束.这时,表示松针的纹元与 Kajiya 表示毛绒的纹元又有明显的不同,因为不同松针的材质不同,方向也不一样,所以需要存储的不但是散射密度 ρ,还有结构场组合 B 中表示松针切线方向 t 以及描述松针局部材质的反射率 K_d 和 K_s.

2.2 纹元的光照模型

对于纹元绘制的光照模型,我们采用类似于 Kajiya 使用的方法.由于松针本身的方向向量(切向量 t)是可以通过纹元的变形精确求取的,因而对双向光照反射函数的计算可以采用局部光照模型的方法.具体地说,松针光照的漫反射分量和镜面反射分量可以用如下光照模型来计算:

$$\Psi_d=K_d\sin(t,l), \quad (3)$$

$$\Psi_s=K_s\cos^n(\theta-\theta')=K_s(\cos\theta\cos\theta'+\sin\theta\sin\theta')^n=K_s[\cos(t,l)\cos(t,e)+\sin(t,l)\sin(t,e)]^n. \quad (4)$$

式中的向量 t,l 和 e 分别是切向量、光线向量和视线向量;K_d 和 K_s 分别是毛发的漫反射系数和镜面反射系数,如图 9 所示.

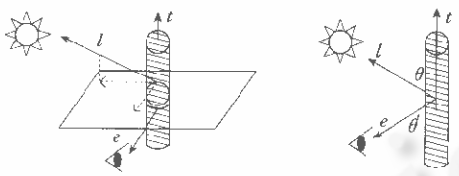

Fig. 9 Calculation of diffuse component Ψ_d(left) and specular component Ψ_s(right)
图9 计算漫反射分量 Ψ_d(左图)和镜面反射分量 Ψ_s(右图)

对于 Kajiya 表示毛绒的纹元,所有绒毛的颜色和材质都是一样的,因此等式(3)和(4)中的 K_d 和 K_s 是常数.而在我们的松针模型中,为了更接近客观事实,不同松针的材质 K_d 和 K_s 是不同的,所以需要将局部材质保存在 Ψ 中.

这样,经过修改的纹元模型就可以构造松针束了.

2.3 方向向量的计算

如前所述,与 Kajiya 的纹元不同,松针的切线方向不是相互平行的,所以需要将切线方向保存在松针纹元数组中.在实际实现过程中,我们记录的是一个在纹元空间中长度小于体元边长二分之一的向量 t_T,如图 10 所示.

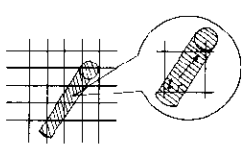
Fig. 10 Tangent vectors in texel array
图10 记录在纹元数组中的切向量

当计算采用点 p 处的方向向量时,假设在纹元空间中,p 点所在体元中心的参考坐标为

$$p_c=\{u_c\ v_c\ w_c\}^T,$$

此体元存储的方向向量为

$$t_T=\{u_t\ v_t\ w_t\}^T,$$

由此可得

$$\boldsymbol{p}_t=\boldsymbol{p}_c+\boldsymbol{t}_T=\{\boldsymbol{u}_c+\boldsymbol{u}_t \quad \boldsymbol{v}_c+\boldsymbol{v}_t \quad \boldsymbol{w}_c+\boldsymbol{w}_t\}^T.$$

我们知道,通过三线性插值可以由参考坐标值求得世界坐标值,假设由 $\boldsymbol{p}_c$ 和 $\boldsymbol{p}_t$ 的参考坐标求得的世界坐标值分别为 $\boldsymbol{P}_1$ 和 $\boldsymbol{P}_2$,那么采样点处的实际方向向量 $\boldsymbol{t}$ 为

$$\boldsymbol{t}=\overrightarrow{P_1P_2}/\|\overrightarrow{P_1P_2}\|.$$

2.4 纹元的光照计算

纹元的具体光照计算过程与科学计算可视化的体绘制技术相似[11,19],参照图 8,在视点处的亮度 B_r 可由下式计算得到:

$$B_r=\sum_{p=P_{near}}^{P_{far}}\left\{\exp\left[-\gamma\sum_{s=P_{near}}^{p}\rho(s)\right]\cdot Ir(p)\right\}. \tag{5}$$

式中

$$Ir(p)=\rho(p)I(p)\cdot[\Psi_d(p)+\Psi_s(p)]+I_a(p)\cdot K_a(p). \tag{6}$$

其中 $\rho(p)$ 为 p 处的散射密度,Ψ_s 和 Ψ_d 可以由式(3)和式(4)求得. $I(p)$ 为到达 p 处的光强度,假设在到达 p 处时,光线穿过 N 个纹元,那么 $I(p)$ 可由下式求得:

$$I(p)=I_0\cdot\exp\left\{-\gamma\left[\sum_{r=p}^{P_{in}}\rho(r)+\sum_{r=P_{out1}}^{P_{in1}}\rho(r)+\sum_{r=P_{out2}}^{P_{in2}}\rho(r)+\ldots+\sum_{r=P_{outN}}^{P_{inN}}\rho(r)\right]\right\}. \tag{7}$$

式中 I_0 为光源的强度.

2.5 绘制结果

图 11 是采用修改后的纹元构造松针束的绘制结果. 光照计算采用的是光线跟踪结合体绘制的方法,从结果图可以看出,这种绘制算法是有效的,它的不足之处在于绘制时间比较长. 以图 11 为例,虽然我们用 BSP 进行场景划分以加快光线跟踪的求交速度,但在 400 Hz Pentium II 的 PC 机上,绘制时间仍然超过一小时.

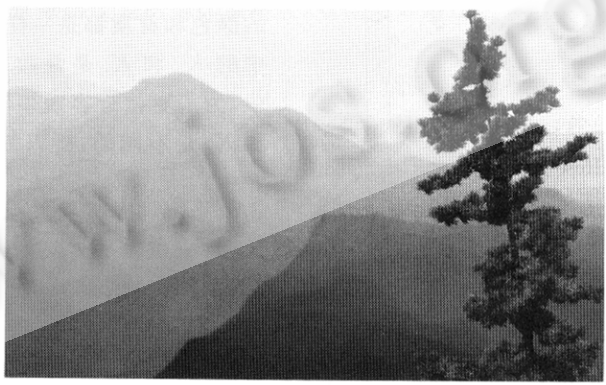

Fig. 11 Rendering result of a pine, represented by texels and polygons

图 11 采用纹元和多边形构造的松树的绘制结果

3 远树的绘制

前面介绍的绘制技术只适合于对近距离的树木的绘制,对于远距离的树,采用上述绘制方法不但计算量大,绘制时间长,而且还很难得到满意的结果,这是因为远距离的树通常数量多. 我们知道,描述一棵树通常需要几万到几百万个多边形,数量巨大的多边形数量不但造成绘制效率的低

下,而且还给存储带来很大的负担.此外,因为一个像素点的颜色值通常是由许多多边形共同作用所产生的,所以将不可避免地出现严重混淆的现象.

对远距离树木的绘制可以大致分为以下 3 种[11]:

(1) 直接绘制基于多边形几何造型的树;

(2) 绘制对原始造型简化了的树木;

(3) 基于三维纹理映射合成.

直接绘制基于多边形几何造型的树木的缺点前面已经提到了,对于数目庞大的原始数据,不但绘制时间长,容易造成明显的混淆,而且存储要求也不是目前一般的硬件所能承受的[8,11,13].为此,我们不考虑采用这种方法.而 Weber 采用的简化原始数据再进行绘制的方法也缺乏高度的真实感[11].我们认为,三维纹理映射(或称体纹理映射)是目前构造和绘制大量远距离树木的有效工具和方法.体纹理的结构与纹元相似,或者说纹元是一种特殊的体纹理结构,因为在纹元包围盒中只有许多细长的圆柱,而体纹理中包含的是许多任意的微面.

体纹理和纹元在结构上的不同,决定了它们的光照模型也不一样.Neyret 提出了构造森林、草地和其他结构复杂、重复出现的物体所构成场景的完整模型[12,13,20].我们就是采用这个模型来构造远树的.然而 Neyret 的模型在构造场景时,体纹理包围盒之间不存在相互穿插,这给造型带来许多不便.其实,体纹理的绘制算法与纹元的绘制算法类似,不同的只是采样点的光照模型,在计算某个像素的颜色时,只要对所有采样点按照它们到视点的距离排序后再作透明度计算和颜色叠加,则体纹理包围盒是可以穿插的.因为篇幅的关系,对体纹理绘制的具体过程就不作介绍了,图 12 就是采用这种算法绘制所得的森林.

Fig. 12 Forest image generated by volume-texture rendering

图 12 用体纹理绘制的远处树林

4 结论和展望

树木的绘制技术一直是计算机图形学绘制技术研究中的一个难题.因为不同类型的场景存在很大的差别,因此采用单一的绘制手段是不可行的.

针对不同的环境特征和不同类型的树木,我们分别采用了基于 OpenGL 的深度缓存阴影生成算法、结合光线跟踪和纹元绘制技术以及结合体绘制技术的光线跟踪方法.从结果图像不难看出,这几种绘制手段对应于各自相应的场景都有效、可行.

显然,草地也可以用纹元来构造,这样,几乎所有的植物场景就都有其对应的绘制手段了.将光线跟踪、纹元绘制和体绘制结合起来,我们就可以绘制包含阔叶树、针叶树和远树的高度复杂的自

然场景,从而生成高度真实感的虚拟自然风景图.

为了提高绘制效率,通常需要将场景进行划分,分别绘制,然后再合成.对划分场景的标准的确定,以及如何合成最终结果、如何考虑不同部分的相互作用,都是值得进一步研究的课题.

References:

[1] Lindenmayer, A. Paracladial system. In: Lindenmayer, A., Rozenberg, G., eds. Automata, Languages, Development. New York: North-Holland Publishing Company, 1976.

[2] Prusinkiewicz, P., Lindenmayer, A. The Algorithmic Beauty of Plants. New York: Springer-Verlag, 1990.

[3] Aono, M., Kunii, T. Botanical tree image generation. IEEE Computer Graphics and Applications, 1984,4(5):10~34.

[4] de Reffye, P., Edelin, C., Francon, J., *et al*. Plant models faithful to botanical structure and development. In: Dill, J., ed. Proceedings of the SIGGRAPH'88 Annual Conference. New York: ACM Press, 1988. 151~158.

[5] Matthew, Holton. Strands, gravity and botanical tree imagery. Computer Graphics Forum, 1994,13(1):57~67.

[6] Oppenheimer, P. E. Real time design and animation of fractal plants and trees. Computer Graphics (SIGGRAPH'86), 1986,20(4):55~64.

[7] Reeves, W. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In: Barsky, B. A., ed. Proceedings of the SIGGRAPH'85 Annual Conference. New York: ACM Press, 1985. 313~322.

[8] Weber, J., Penn, J. Creation and rendering of realistic trees. In: Cook, R., ed. Proceedings of the SIGGRAPH'95 Annual Conference. New York: ACM Press, 1995. 119~127.

[9] Nelson, M., Keiichi, Ohsaki. Rendering trees from precomputed Z-Buffer views. In: Hanrahan, P., Purgathofer, W., eds. Proceedings of the Rendering Techniques'95. New York: Springer-Verlag, 1995. 74~81.

[10] Nelson, M., Curtis, M., Brett, K., *et al*. Plane-Parallel radiance transport for global illumination in vegetation. In: Dorsey, J., Slusallek, P. eds. Proceedings of the Rendering Technique'97. New York: Springer-Verlag, 1997. 239~250.

[11] Chiba, N., Muraoka, K., Doi, A., *et al*. Rendering of forest scenery using 3D textures. The Journal of Visualization and Computer Animation, 1997,8(4):191~199.

[12] Neyret, F. Synthesizing verdant landscapes using volumetric textures. In: Pueyo, X., Schröoder, P., eds. Proceedings of the Rendering Techniques'96. New York: Springer-Verlag, 1996. 215~224, 291.

[13] Neyret, F. Modeling, animating and rendering complex scenes using volumetric textures. IEEE Transactions on Visualization and Computer Graphics, 1998,4(1):55~70.

[14] Watt, A., Watt, M. Advanced Animation and Rendering Techniques. Theory and Practice. New York: ACM Press, 1992.

[15] Williams, L. Casting curved shadows on curved surfaces. Computer Graphics (SIGGRAPH'78), 1978,12(3):270~274.

[16] Reeves, W., Salensin, D., Cook, R. Rendering antialiased shadows with depth maps. In: Stone, M. C., ed. Proceedings of the SIGGRAPH'87 Annual Conference. New York: ACM Press, 1987. 283~291.

[17] Feng, Jin-hui, Chen, Yan-yun, Yan, Tao, *et al*. Going with wind——physically-based animation of trees. Chinese Journal of Computers, 1998,21(9):769~773 (in Chinese).

[18] Wu, En-hua, Chen, Yan-yun, Yan, Tao, *et al*. Reconstruction and physically-based animation of trees from static images. In: Thalmann, M., Thalmann, D., eds. Proceedings of the Eurographics Workshop on Animation and Simulation'99. London: Springer-Verlag, 1999. 157~166.

[19] Kajiya, J., Kay, T. Rendering fur with three dimensional textures. Computer Graphics (SIGGRAPH'89), 1989,23(3): 271~280.

[20] Neyret, F. A general and multiscale model for volumetric textures. In: Davis, W. A., Prusinkiewicz, P., eds. Proceedings of the Graphics Interface'95. Quebec City: Canadian Human-Computer Communications Society, 1995. 83~91.

附中文参考文献:

[17] 冯金辉,陈彦云,严涛,等.树在风中的摇曳——基于动力学的计算机动画.计算机学报,1998,21(9):769~773.

Realistic Rendering of Trees Based on Environmental Features and Species

CHEN Yan-yun[1], YAN Tao[1], ZHANG Xiao-peng[2], WU En-hua[1,3]

[1](*Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*);

[2](*Institute of Automatization, The Chinese Academy of Sciences, Beijing* 100080, *China*);

[3](*Faculty of Science and Technology, University of Macao, Macao, China*)

E-mail: cyy@ox.ios.ac.cn

Received August 9, 1999; accepted October 20, 1999

Abstract: Generation of photo-realistic images of virtual nature scenes is a challenging topic in computer graphics. As a very important part of natural scenery, realistic rendering of trees has attracted much attention. Trees have various types with different shapes and due to their complex structures, realistic modelings, rendering and storage of trees are always a tedious task. To tackle the problem, different rendering strategies ae employed according to different types of trees and different scenes. For broadleaf, the z-buffer shadow generation method is used based on primary functions of OpenGL, for conifer, a ray tracing algorithm including texel-rendering technique is used, and for trees in the far distance, volumetric texture mapping is employed in terms of ray tracing and volume rendering technique. The images produced show that the feasibility and validity of the proposed rendering techniques.

Key words: realistic image synthesis; natural scene, texel; volumetric texture