

基于层次多项式纹理的实时草地绘制*

张淮声¹, 华 炜¹⁺, 余 莉², 鲍虎军¹

¹(浙江大学 CAD&CG 国家重点实验室, 浙江 杭州 310027)

²(安徽师范大学 计算机科学系, 安徽 芜湖 241000)

Layered Polynomial Textures for Real-Time Grassland Rendering

ZHANG Huai-Sheng¹, HUA Wei¹⁺, YU Li², BAO Hu-Jun¹

¹(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China)

²(Department of Computer Science, Anhui Normal University, Wuhu 241000, China)

+ Corresponding author: Phn: +86-571-88206681, Fax: +86-571-88206680, E-mail: huawei@cad.zju.edu.cn

Zhang HS, Hua W, Yu L, Bao HJ. Layered polynomial textures for real-time grassland rendering. *Journal of Software*, 2006,17(Suppl.):85-92. <http://www.jos.org.cn/1000-9825/17/s85.htm>

Abstract: This paper presents a layered polynomial texture (LPT) to represent grass based on its complex geometric structure and diffuses material. LPT is a 5D texture, and its element contains a group of polynomial parameters, to fit the pixels of multiple layers with various lighting directions, and to calculate the actual color in rendering stage. Grass is represented with three LPTs along known view directions. The LPT synthesis method based on wang tile is presented to create tiles, which are mapped onto ground to form grassland. The grassland is rendered in the dynamic lighting condition. Experimental results show that the LPT storage is smaller, and computational cost is lower. Furthermore, LPT could be extended to represent trees with dynamic shadow.

Key words: real-time rendering; grassland; PTM; Wang tile; shadow

摘 要: 根据草的几何结构复杂性和材质的漫反射特性,提出层次多项式纹理(layered polynomial texture,简称LPT)来表示草,可以高效真实地实现这个目标.LPT 是一个五维纹理,每一个纹理单元是一组多项式系数,记录分布在多层像素的不同光照方向下亮度的拟合信息,用于在绘制时计算出纹理像素的颜色.把草表示成 3 个沿已知视线方向的 LPT,然后基于 Wang Tile 的 LPT 合成方法生成拼图块,再映射到地面上得到草地,从而获得动态光照环境下草地的实时绘制.实验结果表明,LPT 的存储量小,运算代价低,而且 LPT 能推广到树的表示,也可以实现树的动态阴影效果.

关键词: 实时绘制;草地;PTM; Wang tile;阴影

实时绘制带阴影效果的自然场景在许多应用中扮演着重要角色,如虚拟现实、三维游戏.草地是自然场景中的一个重要元素,受太阳光和环境光的作用,草的叶子总带有复杂的阴影.由于草的细节过多,并且存在半透明的叶片,因此很难生成随光照方向改变而动态改变的软阴影的实时绘制效果.

* Supported by the National Natural Science Foundation of China under Grant Nos.60021201, 60203014 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312102 (国家重点基础研究发展规划(973))

Received 2006-03-15; Accepted 2006-09-11

对于如何高效真实地绘制自然场景已有过很多研究,对草的多边形模型进行绘制可以得到高细节的绘制效果.Deussen 等人^[1]设计了一个生态模拟系统,包含了地形和各种各样的植物,表现了生动的自然场景.但多边形结构的复杂性,绘制代价比较高,难以实现复杂场景的实时动态阴影效果.Neyret^[2]使用八叉树纹元描述植物,通过光线跟踪算法得到很好的绘制质量,但同样绘制代价很高.Perbet 等人^[3]提出一种对草进行交互的运动方法,该方法使用 3 种模型:三维多边形的草、2.5 维元素和二维纹理,王等人^[4]研究了风吹草动的效果,但是这些方法重点考虑草的运动,没有考虑阴影的绘制问题.基于纹理的方法^[5]来表示草和树,可以有效提高绘制速度,不过仍然没有考虑动态阴影.Meyer 等人^[6]提出一种三维树并带阴影的交互绘制方法.树被转换为层次结构,在绘制过程中计算阴影.但该方法计算量大,目前为止仅应用于树的绘制.

绘制动态光照效果的一种简单方法是根据物体的法向量和光线方向,在 GPU 的像素着色器中应用某种光照明模型来计算,但是难以考虑物体自身的遮挡带来的自阴影,从而降低绘制结果的真实感;阴影图(shadow map)是一种快速有效生成阴影的方法,但是当光线变化时必须更新阴影图,由于植被复杂的几何结构,更新代价非常高,同时半透明的叶子会带来软阴影,阴影图无法处理软影的效果.一些重光照算法已经可以很好地绘制带阴影的物体,如,PRT^[7],BTF^[8],STF^[9].不过这些方法研究的是一般材质的光照特点,且着眼于物体的细节表现.我们的目标是绘制大范围的草地,考虑到大规模场景本身就消耗了大量资源,这要求算法高效而且存储量要小.

草的材质特点是以漫反射为主,并且高光性很弱;形态上的特点是结构复杂,可以看成是多层的复合结构.本文根据草的材质和形态这两方面的特点,提出一种近似但是十分快速的方法 - 层次多项式纹理(layered polynomial texture,简称 LPT),这是一种五维的纹理,每一纹理单元包含一组多项式系数,记录分布在多层像素的不同光照方向下亮度的拟合信息.其拟合思想与多项式纹理映射(PTM)^[10]方法类似,均针对漫反射的材质,但是 PTM 只处理单层纹理,很难描述复杂结构的草,如果把草分割成多层每层使用 PTM,那么数据量和计算量均会线性增长,不利于实时的阴影计算.与 PTM 相比,LPT 降低了数据量,而且层次结构使得绘制草地时能更好地表达细节.我们把草表示成三个方向的 LPT 采样,然后通过基于块的纹理合成方法^[11,12]映射到地面上.为了减少绘制时显存的消耗,我们使用 Wang Tile^[12]的方法预先生成 LPT 拼图块,再非周期地映射到地面上.这样,整个草地的数据量不大,可以一次性装入到显存中进行实时绘制.

1 层次多项式纹理 LPT

1.1 层次多项式纹理函数

PTM 本质上是一个四维的函数表示,给定视线方向和当前光照方向 (l_x, l_y) ,纹理坐标为 (p, q) 的像素点的光亮度为

$$L(p, q; l_x, l_y) = \begin{pmatrix} l_x & l_y & 1 \end{pmatrix} \begin{pmatrix} a_0 & a_2 & a_3 \\ a_2 & a_1 & a_4 \\ a_3 & a_4 & a_5 \end{pmatrix} \begin{pmatrix} l_x \\ l_y \\ 1 \end{pmatrix} = a_0 l_x^2 + a_1 l_y^2 + 2a_2 l_x l_y + 2a_3 l_x + 2a_4 l_y + a_5 \quad (1)$$

其中 a_0, a_1, \dots, a_5 是 PTM 的 6 个系数,通过 SVD 算法分解而得,用来拟合不同光照方向得到的图像.因为 PTM 只能包含一层的信息,并不足以描述复杂的植物形态,所以必须使用多层结构.但如果每一层都使用 PTM,数据量和计算量就会呈线性增长.考虑到绘制时用户观察到的只是离他比较近的层次,远的层次可能被近处的遮挡根本看不到,所以我们重点研究可见的像素点,为此作两个假设:

1) 像素光亮度的传递性.由于植物的叶子是半透明的,某个层次可见像素的光亮度能传递到它后面层次的像素上,所以纹理坐标相同而层次不同的像素有相似的光亮度属性.

2) 每一层次像素的可见性. 植物的叶子通常很密, 存在大量自遮挡, 所以我们认为可见的像素比被遮挡的像素对绘制更重要.

植被的叶片结构复杂, 不同位置的半透明度有可能不一样, 如果对叶片的材质进行精确表示, 则需要建立十分复杂的绘制模型, 不利于快速计算. 考虑到草地等植被的几何具有一定的局部相似性, 因此在一定范围采样得到像素的光亮度属性比较接近, 可认为这些像素满足上述的光亮度传递性, 从而可以使用较为简化的绘制模型进行快速计算, 提高了绘制效率.

我们把位于不同层次而纹理坐标相同的一组像素称为素串(pixseries), 如图 1(a)中灰色点所示. 素串包含 n 个像素, 并分布在各层中, 每个像素有唯一的层次索引. 对于素串, 我们可得到一个多层结构的表示:

$$f(p, q; l_x, l_y, u) = \begin{pmatrix} l_x & l_y & u & 1 \end{pmatrix} \begin{pmatrix} b_0 & b_3 & b_4 & b_6 \\ b_3 & b_1 & b_5 & b_7 \\ b_4 & b_5 & b_2 & b_8 \\ b_6 & b_7 & b_8 & b_9 \end{pmatrix} \begin{pmatrix} l_x \\ l_y \\ u \\ 1 \end{pmatrix} \quad (2)$$

这是一个五维函数, 其中 u 等于素串中某个像素的层次索引除以 n . 注意到如果层数 n 为 1, 则 u 为 0, 公式(2)退化为公式(1), 所以我们的方法比 PTM 更一般, 考虑了多层的拟合信息, 称为层次多项式纹理函数, 该函数有 10 个系数(b_0, b_1, \dots, b_9).

我们需要通过判断每个像素的可见性来计算 LPT 的系数. 然而从某个视线方向可见的像素很可能从另一个角度观察被遮挡, 因此, 我们只在视域空间 Ω 中估算可见率, 视域空间 Ω 一般为半球或立方体. 由于在整个视域空间计算可见率仍然比较困难, 我们只选择 m 个方向计算近似值:

$$\delta_i = \int_{\theta \in \Omega} \text{Vis}(Pix_i, \theta) d\theta \approx \frac{1}{m} \sum_{\theta=0}^m \text{Vis}(Pix_i, \theta) \quad (3)$$

其中 Pix_i 是素串中第 i 个像素, 其可见率为 δ_i . 函数 $\text{Vis}(Pix_i, \theta)$ 表示像素 Pix_i 在视线方向 θ 观察, 该像素是否可见, 如图 1(b)所示. 可见率 δ_i 的取值范围为 0~1 之间, 例如图 1(b)中, 该像素可见率为 2/5.

为了得到素串的系数(b_0, b_1, \dots, b_9), 我们需要求解一个关于不同采样方向, 素串亮度的最小化问题:

$$\min \sum_{i=0}^n \delta_i \left(\sum_{l_x} \sum_{l_y} (f(p, q; l_x, l_y, u_i) - Lum_i(p, q; l_x, l_y))^2 \right) \quad (4)$$

其中 u_i 是素串中第 i 个像素的 u 值, Lum_i 为该像素在光照方向(l_x, l_y)下的亮度, 可通过预先采样得到. 该最小化问题可利用线性最小二乘法求解.

同一素串中每个像素都具有相同的 LPT 系数(b_0, b_1, \dots, b_9), 此外, 每个像素还保存着它的原始颜色信息(R, G, B), $i=0, 1, \dots, n-1$. 像素原始颜色可以在没有光照的条件下对模型采样得到. 这样素串的总存储量为 $10s + 3n$ 字节, 其中 s 是每个系数的数据大小. 而 PTM 的存储量为 $6sn + 3n$ 字节, 所以我们的 LPT 减少了 $(6n-10)s$. 只要层次数大于 2, 节约的存储空间就很可能可观了. 我们在实验中最少使用了 6 个层次. 最后, 我们把计算得到的亮度值与像素的原始颜色值相乘, 就得到在光源照射下的最终颜色.

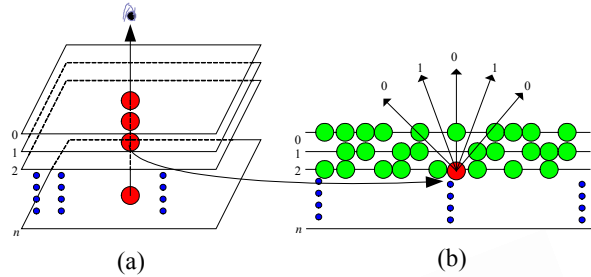


Fig.1 Multi-Layer structure of LPT. (a) There are n layers, and the gray dots form a pixseries. (b) For each pixel of the pixseries, its visible ratio is evaluated through the pre-selected directions

图 1 LPT 的多层结构.(a) n 个层次的结构, 灰色点构成一个素串.(b) 素串中的每一个像素的可见率由预先选择的方向计算得到

1.2 基于GPU的加速算法

为了得到动态光照下的绘制效果,必须对采样纹理的每个像素逐一计算,那么直接利用公式(2),计算量将会很大,难以达到实时的绘制效果.考虑到目前 GPU 发展迅猛,尤其在数据的并行处理上具有优势,可以加快绘制速度,我们可以利用 GPU 进行加速.

如果直接在 GPU 中使用式(2)计算,计算量要比(1)大.我们通过分析发现,对于某个光线方向 (l_x, l_y) ,所有层的数据仅依赖于系数 u ,所以式(2)转化为

$$\begin{cases} A = \begin{pmatrix} l_x & l_y & 1 \end{pmatrix} \begin{pmatrix} b_0 & b_3 & b_6 \\ b_3 & b_1 & b_7 \\ b_6 & b_7 & b_9 \end{pmatrix} \begin{pmatrix} l_x \\ l_y \\ 1 \end{pmatrix} \\ B = 2(b_4 \ b_5 \ b_8)(l_x \ l_y \ 1)^T \\ C = b_2 \end{cases} \quad (5)$$

这里 A 和 B 可以先计算出来,然后再用它们计算每个像素的结果,第 i 层像素的亮度值可表示为

$$f_u = A + Bu + Cu^2 \quad (6)$$

这样利用 GPU 的计算过程可以分为两步:第 1 步,根据某个光线方向 (l_x, l_y) ,计算每个素串的 A 和 B ,并以纹理方式保存;第 2 步,利用式(6)计算素串中每个像素的实际亮度值,并乘以原始颜色得到最终结果.对于每个素串,如果层数为 n ,则如果使用 PTM,计算量为 $5n$ 次乘法和 $5n$ 次加法,而使用 LPT 的计算量仅为 $5+2+2n=2n+7$ 次乘法和 $2n+7$ 次加法.如果 n 大于 3,显然 LPT 的计算量比 PTM 小.

为了方便 GPU 的计算,我们需要把 LPT 的系数组装成纹理的形式,并预先存储在显存中.因为系数 (b_0, b_1, \dots, b_9) 是浮点类型,所以每个系数 b_i 需要先被量化到 0~255 之间,得到 b'_i ,如式(7)所示

$$b'_i = S_i(b_i + M_i) \quad i = 0, 1, \dots, 9 \quad (7)$$

其中 S_i 和 M_i 分别表示缩放值和基值, b_i 的值可以由公式(7)反求得到.这样就可以使用一个字节表示每个系数,每张纹理有 RGBA 四个通道,则共需要 3 张纹理存储整个 LPT 系数.在绘制阶段,LPT 系数和原始颜色均以纹理方式传送给 GPU,缩放值和基值也作为常参数传入.

2 基于 LPT 的草地生成

2.1 构造草的Slice和Blade

为了能够表现草地的多方向绘制效果,便于利用 LPT 来计算动态光照,我们对草的模型进行 3 个方向的采样,并建立 LPT 结构:一个沿垂直的视线方向观察得到,称为 Slice;另两个沿水平视线方向观察得到,称为 Blade,如图 2(a)所示.

为了得到 Slice,我们沿着垂直的方向观察草的多边形模型,用成对的切平面将模型切分成若干个剖分层.对每一个剖分层,沿不同的光照方向通过光线跟踪算法进行采样,得到一组图像(图 2(b)),每张图像都包含 RGBA 4 个通道.如果 Slice 的数量为 N_s ,预处理的光线方向为 N_l ,则采样的图像数量为 $N_s \times N_l$ (图 2(c)).完成采样后,再通过公式(4)计算出 Slice 的 LPT 系数(图 2(d)).两个 Blade 也按照同样方法处理得到.

当从草的上方观察时,Slice 看起来效果不错.不过如果观察方向接近水平,例如在草地上行走方式进行观察,Slice 的效果则难以令人接受,此时必须依赖 Blade.Lengyel 等人^[13]使用一种垂直采样纹理(称为 Fin)来提高皮毛的绘制效果.本文的 Blade 与 Fin 类似,不同的是 Blade 包含了多层信息,并且记录了 LPT 数据,从而可以实时绘制带有阴影的草.

2.2 LPT拼图块的草地构造

本节介绍基于 Wang Tile 的 LPT 拼图块方法,该方法可确保预先设计好的拼图块沿边界进行无缝连接.我们先从采样的草数据分别构造出 Slice 拼图块和 Blade 拼图块,再把它们映射到地面上.

我们从草的 Slice 中选择几个大小相同的菱形块用于构造 Slice 拼图块,这些菱形块同样也是多层结构.4 个菱形块被结合在一起,通过图像缝合法^[12,14]来搜索分割路径,这样就构成一个 Slice 拼图集.与文献[14]不同的是,我们的 Slice 拼图块的基本单位是包含 LPT 信息的素串,因此我们需要综合考虑 LPT 系数和原始颜色来搜索分割路径.

计算分割路径的关键是比较两个不同素串的差异,如果菱形块有 n 层,一个素串就有 10 个系数和 n 个原始颜色.我们根据 LPT 系数对两个素串进行比较,并求出误差.首先,计算素串中 n 个像素原始颜色的均值,记为 $(\bar{R}, \bar{G}, \bar{B})$.设两个素串 i 和 j 的原始颜色均值分别为 $(\bar{R}_i, \bar{G}_i, \bar{B}_i)$ 和 $(\bar{R}_j, \bar{G}_j, \bar{B}_j)$,LPT 函数分别为 $f_i(l_x, l_y, u)$ 和 $f_j(l_x, l_y, u)$,则误差 $\varepsilon(i, j)$ 为

$$\varepsilon(i, j) = E(\bar{R}_i, \bar{R}_j) + E(\bar{G}_i, \bar{G}_j) + E(\bar{B}_i, \bar{B}_j) \quad (8)$$

因为素串颜色随光线方向的变化而变化,所以我们需要在光照空间计算误差, $E(\bar{R}_i, \bar{R}_j)$ 可以通过下式计算:

$$E(\bar{R}_i, \bar{R}_j) = \iiint_{l_x, l_y, u} \|\bar{R}_i \cdot f_i(l_x, l_y, u) - \bar{R}_j \cdot f_j(l_x, l_y, u)\|^2 dl_x dl_y du \quad (9)$$

其中 $-1 \leq l_x, l_y \leq 1, 0 \leq u \leq 1$, $E(\bar{G}_i, \bar{G}_j)$ 和 $E(\bar{B}_i, \bar{B}_j)$ 类似公式(9),不再列出.

从 4 个菱形块构造出的 Slice 拼图块被存储成一个 LPT 映射图.为了保证绘制过程中拼图块能无缝地贴在地面上,要注意存放在 LPT 映射图里的拼图块必须是沿边界连续的.因为 Blade 是和 Slice 绑定在一起,所以在确定好 Slice 的拼图块后,只需要把对应的 Blade 块取出即可.

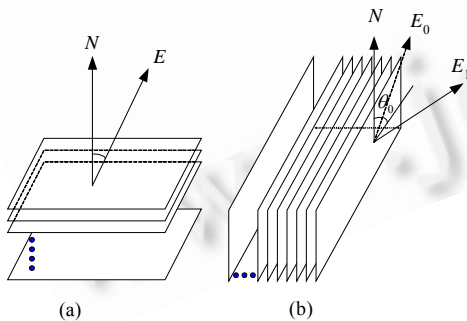


Fig.3 Two blending scheme. (a) ω_s is decided by the normal and the viewing direction, (b) ω_b is calculated,

θ_0 is a presetting parameter

图3 两种融合方法.(a)由法向和视线方向计算的 ω_s , (b) ω_b 的计算, θ_0 为预设阈值

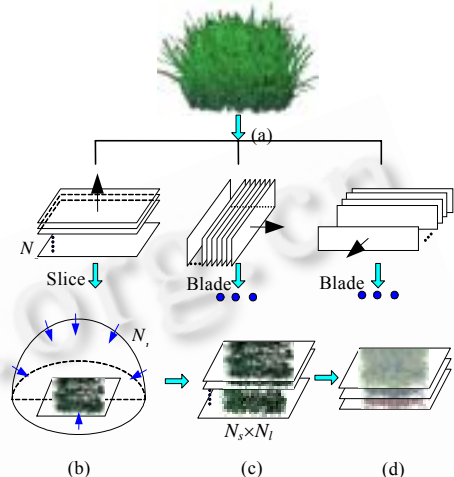


Fig.2 Sampling procedure of the grass

图2 草的采样流程

地面被分割成许多大小相同的正方形,按照从左到右、从下到上的顺序选择拼图块进行映射.我们根据邻接边来选择合适的 Slice 拼图块,如果有多个满足条件的拼图块,则随机选择一个.附加在 Slice 拼图块上的 Blade 拼图块也随之映射.为减少存储量,我们仅存储拼图块的索引值,在绘制过程中根据索引计算 Slice 拼图块和 Blade 拼图块的纹理坐标.

当使用 LPT 拼图块拼接草地场景时,由于其相邻块的邻接部分在草的 Slice 和 Blade 中处于连续的位置,所以保证了邻接区域的可见性也是连续的,不需要重新计算这些像素的可见性.但是在构造 LPT 拼图块进行缝合操作时,其缝合路径上像素的可见性可能会变化,因此比较理想的方案是重新计算这些像素的可见性和 LPT 系数,但这将增大预处理的复杂性.从实验结果来看,这些像素的数量很少,其可见性的变化对绘制质量影响不大.因此我们没有做进一步的优化.

2.3 使用LPT绘制草地

绘制一块草的过程为,先根据当前的光线方向利用 LPT 计算出 Slice 和 Blade 的颜色值,再用纹理映射技术在融合模式下绘制它们.为了使 Slice 和 Blade 随视线改变而平滑过渡,在绘制每一帧前,需要根据视线方向计算融合的权值.

草的法向垂直于 Slice,平行于 Blade,记为 N ,如图 3 所示,视线方向为 E .Slice 的融合权值为: $\omega_s = \max(N \cdot E, 0)$.当观察者以垂直角度观察草的时候, ω_s 为 1;以水平角度观察时, ω_s 为 0.

为了增加融合过程的真实度,只有当 N 和 E 的夹角大于阈值 θ_0 时,Blade 才可见.如图 3(b)所示,当视线方向为 E_0 时 Blade 不可见,当为 E_1 时 Blade 可见.因此 Blade 的融合权值为 $\omega_b = \max\left(1 - \frac{|N \cdot E|}{\cos(\theta_0)}, 0\right)$.

我们先用 ω_b 绘制所有的 Blade,再用 ω_s 绘制所有的 Slice,就能得到 Slice 和 Blade 的平滑过渡.草地的绘制过程和草的绘制过程一样.

3 实现和实验结果

我们实现了基于层次多项式纹理的草地绘制算法,统计的数据是在一台 CPU 主频为 2.4GHZ、内存为 1GB、显卡为 GeForce FX 6800 GT 的计算机上测得.

测试使用的草模型约包含 2 万个几何三角形,我们设置了 15 个光线方向,均匀分布在采样的上半球面上.对 Slice 和 Blade 的每一个层次,沿 15 个不同的光照方向可采样得到 15 幅图像.由于 Blade 和 Slice 的结构是一样的,我们只测试了 Slice 的性能.表 1 中,PTM 的存储量随分辨率和层数的增加呈线性增长,而 LPT 的存储量只和分辨率相关,与层数无关.PTM 和 LPT 的存储量的比率列于表中第 6 列.明显,LPT 比 PTM 所占存储空间存储低许多.

Table 1 Storages of slices for PTM and LPT

表 1 PTM 和 LPT 的 Slice 的存储量

纹理分辨率	层数	存储容量(MB)			
		原始颜色纹理	PTM	LPT	LPT: PTM
256×256	8	2	4.89	2.93	0.60
256×256	16	4	9.64	2.93	0.31
512×512	16	16	41.8	11.73	0.28

虽然 LPT 使用了比 PTM 多的系数,但应用 2.2 节的 GPU 加速算法后,绘制性能并没有降低.对 16 层、分辨率为 256×256 的 Slice,使用 PTM 的帧速约 3-5fps(帧/秒),使用第 2.2 节的 GPU 加速算法后,LPT 的绘制时间不到 10 毫秒,帧速为 200fps 左右,即使对于 512×512 的 Slice,帧速也可以达到 70fps.

草的绘制结果如图 4 所示,我们采用的是 16 个层次的 Slice 和 8 个层次的 Blade,分辨率分别为 256×256 和 512×128,达到了实时性能.为了提高绘制质量,我们也使用 LPT 方法捕捉草在地面上的投射阴影.

图 4(a),(d)是 PTM 方法的绘制效果,图 4(b),(e)是 LPT 方法的绘制效果.因为 LPT 考虑了像素的可见性来计算像素的光亮度,所以 LPT 的绘制结果和 PTM 很接近.图 4(c),(f)是 LPT 在另一光照方向下的绘制结果.实验中设置的融合阈值 θ_0 为 30.在我们的 Video 演示中,Slice 和 Blade 是渐进方式切换的,这与真实的草并不一样,但动态光照下草地的实时绘制性能还是很吸引人的,并很实用.

由于树木的材质是以漫反射为主,其结构也可以看成是层次的表示,所以把 LPT 引入到我们前期的工作——层次深度拼图集(layered depth mosaic,简称 LDM)^[15]中,从而能够绘制动态光照下的树木.具体过程是,首先在树的包围球上选择若干个采样点(文献[15]中,我们选择了 15 个采样点),在每个采样点对树木通过光线跟踪算法进行采样,可得到一组包含深度信息的多层图像,图像的每个像素拥有多个光照方向下的颜色值,然后按照本文的方法构造每个采样点的 LPT 纹理.接着对这些 LPT 纹理进行分块和纹理压缩操作,并根据深度信息确定每个分块的空间位置(具体过程可参看文献[15]),得到了树木的 LPT 绘制数据.在绘制时,可根据视点信息选择多个采样点对应的 LPT 数据,以纹理融合方式进行处理,并通过 GPU 计算出纹理的颜色值.图 5 显示了两棵树木

在不同光照下的绘制结果,其阴影随着光照方向的变化而变化,绘制时间为 10ms 左右.

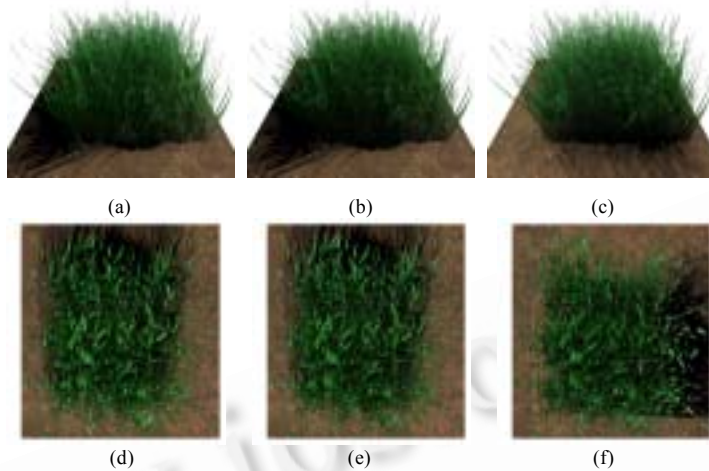


Fig.4 Rendering results of grass. (a), (d) is rendered by PTM, and (b),(c),(e),(f) by LPT

图 4 草地的绘制效果.(a),(d)是使用 PTM 方法,(b),(c),(e),(f)通过 LPT 方法



Fig.5 Real-Time rendering results of the trees in different lighting directions respectively

图 5 动态光照下不同树木的实时绘制效果

在绘制草地方面,我们使用了 18 个 Slice 拼图块来构造草地,每个 Slice 拼图块的分辨率为 64×64.图 6(a)是 100×100 个拼图块的草地,其帧速约为 50fps.图 6(b)是有 200×200 拼图块的草地和 500 棵树的真实场景,帧速约为 25fps.通过 LPT 方法和 GPU 加速,我们可以实时绘制包含动态阴影的自然场景.可通过网络获得相关的视频演示: <http://www.cad.zju.edu.cn/home/zhs/lpt/index.html>.

我们方法的不足之处在于,LPT 是一种基于纹理的方法,利用它来表示和绘制草地,虽然绘制速度很快,但是无法高质量地表示草的形态,此外对于光照阴影采用的拟合方法,会有一些的能量损失.所以我们的方法主要适用于草地的整体绘制效果.



(a) The grassland with 100×100 tiles
(a) 100×100 个拼图块的草地



(b) A natural scene including the grassland with 200×200 tiles and trees
(b) 一个自然场景,包含 200×200 个草地的拼图块和 500 棵树

Fig6 Rendering results of natural landscape

图 6 草地的绘制效果

4 结论和展望

本文提出的层次多项式纹理(LPT)算法,可实时绘制带光照条件的草和树.不同于 PTM,LPT 重点研究多层次结构,适于描述自然植物,并且其存储量和计算代价都不大.LPT 不是一种精确的绘制算法,主要适用于漫反射为主的物体.但对于植物来说,主要是漫反射起作用,且镜面反射很少,所以利用 LPT 能绘制自然场景的动态阴影,整体的绘制效果也令人满意.

为了把草映射到地面上,我们把草的采样分解成基于 LPT 的 Slice 和 Blade 结构,并基于 WangTile 算法,提出了 LPT 的地面拼图构造算法.我们还利用 LPT 实现了带光照环境下的树的绘制.我们目前的工作还不能用于动态的自然场景模拟,在下面的研究中,需要考虑随风运动的草地绘制.此外,把 LPT 推广到其他领域,如动态光照下的动物皮毛绘制,将是件有意义的工作.

References:

- [1] Deussen O, Hanrahan P, Lintermann B, Mëch R, Pharr M, Prusinkiewicz P. Realistic modeling and rendering of plant ecosystems. In: Hefley W, ed. Proc. of the SIGGRAPH'98. New York: ACM Press, 1998. 275-286.
- [2] Neyret F. Synthesizing verdant landscapes using volumetric textures. In: Pueyo X, Schröder P, eds. Eurographics Workshop on Rendering'96. London: Springer-Verlag, 1996. 215-224.
- [3] Perbet F, Cani MP. Animating prairies in real-time. In: Hughes JF, Séquin CH, eds. ACM Interactive 3D Graphics. New York: ACM Press, 2001. 103-110.
- [4] Wang CB, Wang ZY, Zhou Q, Song CF, Guan Y, Peng QS. Dynamic modeling and rendering of grass wagging in wind. Computer Animation and Virtual Worlds, 2005,16(3,4):377-389.
- [5] Reche A, Martin I, Drettakis G. Volumetric reconstruction and interactive rendering of trees from photographs. ACM Trans. on Graphics, 2004,23(3):720-727.
- [6] Meyer A, Neyret F, Poulin P. Interactive rendering of trees with shading and shadows. In: Gortler SJ, Myszkowski K, eds. Eurographics Workshop on Rendering 2001. London: Springer-Verlag, 2001. 183-196.
- [7] Sloan PP, Kautz J, Snyder J. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency light environments. ACM Trans. on Graphics, 2002,21(3):527-536.
- [8] Liu XG, Yu YZ, Shum HY. Synthesizing bidirectional texture functions for real-world surfaces. In: Campbell M, ed. Proc. of the SIGGRAPH 2001. New York: ACM Press, 2001. 97-106.
- [9] Chen YY, Tong X, Wang JP, Lin S, Guo BN, Shum HY. Shell texture functions. ACM Trans. on Graphics, 2004,23(3):343-353.
- [10] Malzbender T, Gelb D, Wolters H. Polynomial texture maps. In: Campbell M, ed. Proc. of the SIGGRAPH 2001. New York: ACM Press, 2001. 519-528.
- [11] Decaudin P, Neyret F. Rendering forest scenes in real-time. In: Fellner D, Spencer S, eds. Eurographics Symp. on Rendering 2004. Aire-la-Ville (Switzerland): Eurographics Association, 2004. 93-102.
- [12] Cohen MF, Shade J, Hiller S, Deussen O. Wang tiles for image and texture generation. ACM Trans. on Graphics, 2003,22(3):287-294.
- [13] Lengyel J, Praun E, Finkelstein A, Hoppe H. Real-Time fur over arbitrary surfaces. In: Hughes JF, Séquin CH, eds. ACM Symp. on Interactive 3D Graphics 2001. New York: ACM Press, 2001. 227-232.
- [14] Efros AA, Freeman WT. Image quilting for texture synthesis and transfer. In: Campbell M, ed. Proc. of the SIGGRAPH 2001. New York: ACM Press, 2001. 341-346.
- [15] Zhang HS, Hua W, Wang Q, Bao HJ. LDM-Pack: A new method for rendering trees rapidly. Journal of Image and Graphics, 2004, 9(10):1216-1222 (in Chinese with English abstract).

附中文参考文献:

- [15] 张淮声,华炜,王青,鲍虎军.层次深度拼图集:一种新的树木快速绘制方法.中国图像图形学报,2004,9(10):1216-1222.



张淮声(1976 -),男,安徽六安人,博士生,主要研究领域为计算机图形学,实时绘制,自然场景模拟.



余莉(1976 -),女,讲师,主要研究领域为数字几何处理,隐式曲面.



华炜(1972 -),男,博士,副研究员,主要研究领域为实时模拟和绘制系统,虚拟现实和软件工程.



鲍虎军(1966 -),博士,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,数字几何理论.