

## 真实感动态树场景的研究与实现

崔翔\*, 姜小峰

(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

(\*通信作者电子邮箱 medivhc@gmail.com)

**摘要:** 树木的动态显示在自然场景模拟中有着重要的地位。采用 Cook-Torrance 光照模型和预计算的半透明贴图实现了叶片的绘制, 使用梁模型导出的拟合多项式结合偏移修正实现枝干的快速形变计算, 通过索引枝干层级数据贴图实现 GPU 中多级枝干的形变。预计算和 GPU 的使用使得模拟过程中真实感和实时性得以平衡。实验结果表明, 该方法实现了快速真实的动态树场景渲染。

**关键词:** 树场景; 叶片渲染; 半透明; 物理动画; 图形处理器

**中图分类号:** TP391.9 **文献标志码:** A

### Research and implementation of realistic dynamic tree scene

CUI Xiang\*, JIANG Xiaofeng

(School of Computer Science and Technology, Soochow University, Suzhou Jiangsu 215006, China)

**Abstract:** Dynamic tree rendering plays an important role in the natural scenery simulation. In this paper, by using Cook-Torrance lighting model and pre-computed translucency texture, rendering scattering and translucency of the leaf were implemented. Using the polynomial fitted from tapered circular beam model expression and length correct method, the speed of calculation deform was boosted. By introducing the hierarchical branches texture with index, branches deform could be calculated in Graphic Processing Unit (GPU). Using pre-compaction and GPU helps to balance the reality and real-time in the simulation. The experiments show that the proposed method can render the dynamic tree scene vividly and rapidly.

**Key words:** tree scene; leaf rendering; translucency; physical animation; Graphics Processing Unit (GPU)

## 0 引言

树木是自然场景中重要的组成部分, 具有真实感的树木动态模拟在虚拟现实相关领域中起着举足轻重的作用, 其应用范围包括动画、游戏和军事等领域。因此对树场景动态实时模拟技术的研究一直以来都是一个热点和难点。

目前, 对于树木的绘制可分为基于图像的方法和基于几何的方法。基于图像的方法极大地降低了树模型的面片数量并对光照和阴影进行了预计算, 效率高, 但在近距离观察时效果不理想, 多用于大规模的场景区或背景的绘制<sup>[1]</sup>。基于几何的绘制为普遍使用的方法, Sousa<sup>[2]</sup>使用了叶片的厚度贴图提升了树木的显示效果。Kharlamov 等<sup>[3]</sup>在计算叶片的时候考虑了叶片相互遮挡产生的自阴影的效果。但这些方法在增强真实感的同时引入了大量的计算。

对于树木枝干受风力作用的形变及动画, 可以分为过程式和基于物理两类方法。过程式方法一般不考虑树的物理属性, 采用经验公式计算树在风力作用下的位移, 虽然效率较高, 但不够精确。Zioma 等<sup>[4]</sup>使用了刚性梁绕轴周期摆动的方式实现动画, 虽然速度快, 但动画显得重复单调。对于基于物理的方式, 李占利等<sup>[5]</sup>将枝干分段, 使之作为连接的弹簧模型, 计算每段受风力产生的偏移。Hu 等<sup>[6]</sup>同样使用分段的方式, 同时加入了枝干沿轴的扭转。该类方法中每个分段节点的偏移必须逐一计算并累加, 精确模拟需要增加分段, 导致计算量较大, 且难以并行。Yang 等<sup>[7]</sup>使用层次矩阵结构, 使用统一计算构架 (Compute Unified Device Architecture, CUDA)

实现了枝干的形变, 但 CUDA 平台使得使用该方法存在一定的局限性。

本文使用 Cook-Torrance 光照模型和半透明贴图的方式, 增加了较少的运算量, 展现了树木的叶片的光照细节。通过 Euler-Bernoulli 梁模型得到的挠度多项式计算偏移, 并使用长度修正的方式得到最终顶点的位置。克服了以往树木形变方法计算速度较慢的缺点, 所获场景在拥有良好局部细节的情况下达到了实时性的要求。

## 1 叶片光照

由于叶片有着复杂的光线传播特性, 常规的 Phong 模型很难体现出其半透明特性以及次表面反射的特性。本文基于文献[8], 通过使用 Cook-Torrance 局部照明模型及半透明贴图高效地实现了叶片在不同太阳光角度下的散射、半透明等效果。

### 1.1 叶片光照计算步骤

叶片光照可由以下公式表示:

$$L = L_a + L_{rs} + L_{rd} + L_t \quad (1)$$

其中:  $L_a$  为环境光项、 $L_{rd}$  为漫反射项,  $L_{rs}$  和  $L_t$  为高光反射项和透射项。

为了真实地反映叶片与太阳的光线交互。叶片的正面和背面分别使用了不同的贴图及参数。如图 1 所示, 叶片的光照共分四种情况。

式(1)中的高光反射项和透射项只有一项不为零。对于图 1(a)、(b)的情况, 即视线与光源同侧时,  $L_t$  为 0。对于图 1

收稿日期: 2012-12-17; 修回日期: 2013-01-29。

作者简介: 崔翔 (1988-), 男, 江苏徐州人, 硕士研究生, 主要研究方向: 计算机图形学、虚拟现实; 姜小峰 (1972-), 男, 江苏丹阳人, 副教授, 主要研究方向: 计算机图形学、虚拟现实。

中(c)、(d)的情况,即视线与光源异侧时, $L_{rs}$ 为0。

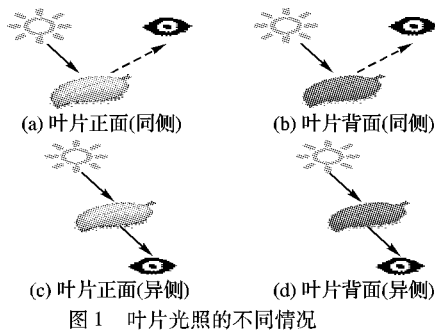


图1 叶片光照的不同情况

## 1.2 Cook-Torrance 光照模型

Cook-Torrance 光照模型由 Cook 等<sup>[9]</sup>提出,在给定法向量  $N$ , 视线方向  $V$ , 入射光方向  $L$  时有如下公式:

$$R_s = \frac{F * D * G}{(N \cdot V) * (N \cdot L)} \quad (2)$$

其中: $F$  为 Fresnel 反射系数, $D$  为微平面分布函数, $G$  为几何衰减系数。

Fresnel 反射系数  $F$  表示反射方向上的光强占原始光强的比率,有:

$$F = f_0 + (1 - f_0)(1 - V \cdot H)^5 \quad (3)$$

其中: $f_0$  为入射角接近 0 时候的 Fresnel 反射系数, $H$  为半角向量。

微平面分布函数  $D$  常用 Backmann 分布函数表示:

$$D = \frac{1}{m^2 \cos^4 \alpha} e^{-\frac{\cos^2 \alpha - 1}{m^2 \cos^2 \alpha}} \quad (4)$$

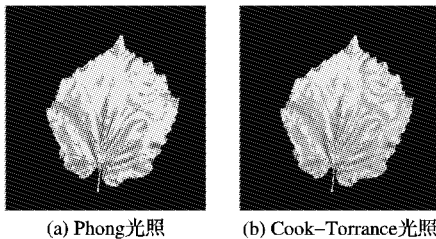
其中: $\alpha$  为是顶点法向量  $N$  和半角向量  $H$  的夹角; $m$  用于度量表面的粗糙程度,较大的  $m$  值对应于粗糙平面,较小的  $m$  值对应于较光滑的表面。

几何衰减系数  $G$  的定义为:到达观察者的光的最小强度。可由如下公式表示:

$$G = \min(1, G_1, G_2) \quad (5)$$

其中  $G_1 = \frac{2(N \cdot H)(N \cdot L)}{V \cdot H}$ ,  $G_2 = \frac{2(N \cdot H)(N \cdot V)}{V \cdot H}$ 。

图2为结合法线贴图分别使用 Phong 光照和 Cook-Torrance 光照单独渲染的叶片效果图,从中可以看出 Phong 光照模型的高光效果过于艺术化, Cook-Torrance 则显得更加柔和、有质感,适合表现叶片光照的细节。



(a) Phong光照 (b) Cook-Torrance光照  
图2 不同光照模型渲染单片叶片的对比

## 1.3 反射项的计算

式(1)中  $L_{rd}$  的计算采用了 1.2 节介绍的 Cook-Torrance 光照模型。由于叶子的正反表面光滑程度不一,除了使用不同的颜色贴图及法线贴图,在使用式(4)时,对应树叶的不同表面,使用了不同大小的  $m$  值。使用较小的  $m$  值对应叶的正面,使用较大的  $m$  值对应叶的背面。

## 1.4 半透明项的计算

为了快速计算半透明项,本文将叶片  $x_0$  处的透光率  $\rho_t$  以及半透明向量  $h$  分别存储为贴图。 $h$  由数据采集设备获取逐

像素反射率、透光率、厚度和法线等信息后生成<sup>[8]</sup>。在渲染时,沿给定  $\omega_D$  方向上,单位入射光强度  $L_D$  下,  $x_0$  处的半透明项为:

$$L_{t,rec}(x_0, \omega_D) = L_D \rho_t(x_0) \sum_{i=1,3} h_i(x_0) \sqrt{\frac{3}{2\pi}} H_i \cdot \omega_D \quad (6)$$

其中  $H_i$  为 Half Life2 基向量。

最终叶片的渲染效果如图3所示,在没有增加过多运算的情况下,很好地表现了叶子正反面在不同光照下的效果。



图3 叶片渲染的细节

## 2 枝干的形变模拟

本文将分层的枝干结构数据存储到贴图中,并赋予索引,以便在 GPU 中计算枝干的层次形变。使用拟合后的多项式结合长度修正的方式实现了枝干形变的快速计算。

### 2.1 枝干的层次表示

为了能够并行计算每个枝干的形变,避免计算结果间的依赖,采用了如下的策略:

首先,使用一个贴图记录所有层次的枝干的信息,这些信息包括枝干的坐标转换基向量和枝干长度等,并为每个枝干指定一个索引来进行贴图的查找。同个枝干上的每个顶点拥有相同的索引。对于每个顶点,需要记录其所在枝干的索引以及所有父层次级别的索引。

其次,通过使用归一化的本地坐标系  $x \in [0,1]$  来描述不同兄弟枝干在相同父枝干上的位置关系。在初始化的阶段对于每个顶点指定了各级的相对位置信息。图4表示了层次为3层的各个顶点所需要记录的相对位置信息。

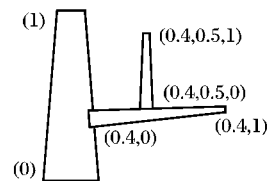


图4 3层枝干的层次表示

### 2.2 Euler-Bernoulli 梁模型

Euler-Bernoulli 梁模型是一种适用于长度半径比较大的结构力学模型,是一种非均匀梁模型,因此能很好地模拟自然界中树木普遍存在的底部较粗而顶端较细的特性<sup>[10-11]</sup>。该模型可用如下公式表示:

$$\frac{\partial^2}{\partial x^2} \left( EI(x) \frac{\partial^2 u(x)}{\partial x^2} \right) = F \quad (7)$$

其中: $u(x)$  是根据常量  $F$  得到的挠度值,  $I$  是截面惯性矩,  $E$  是弹性模量。

每个枝干底端为固定端,其边界条件为:

$$\begin{cases} u|_{x=0} = 0 \\ \frac{\partial u}{\partial x}|_{x=0} = 0 \end{cases} \quad (8)$$

自由端的边界条件为:

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} \Big|_{x=L} = 0 \\ \frac{\partial^3 u}{\partial x^3} \Big|_{x=L} = 0 \end{cases} \quad (9)$$

为了简化计算,将梁模型规范化为单位长度,并用锥度比  $\alpha$  表示树枝两端的半径比。

$$\begin{cases} r_{1,2} = \frac{s_{1,2}}{L} \\ \alpha = \frac{r_2}{r_1} \end{cases} \quad (10)$$

梁的轴向的截面惯性矩为:

$$I = \frac{\pi r^4}{4} \quad (11)$$

由式(10)和式(11)得在  $x$  位置的截面惯性矩为:

$$I(x) = \frac{\pi r_1^4 ((\alpha - 1)x + 1)^4}{4} \quad (12)$$

在给定的边界条件下方程(7)的解析式为:

$$u(x) = \frac{E'F}{r_1^4} (x(\alpha - 1)(6 + x(\alpha - 1)(2x(\alpha - 1)(3 + (\alpha - 3)\alpha) + 3(4 + (\alpha - 2)\alpha))) - 6(1 + x(\alpha - 1))^2 \ln(1 + x(\alpha - 1))) \cdot (3\pi(1 + x(\alpha - 1))^2(\alpha - 1)^4)^{-1} \quad (13)$$

式(13)可通过最小二乘法拟合为多项式(其中  $E'F/r_1^4$  已标准化):

$$u(x) = c_2 x^2 + c_4 x^4 \quad (14)$$

$c_2, c_4$  系数由表1所示,其中  $\Delta x_{\max}$  为最大误差值。

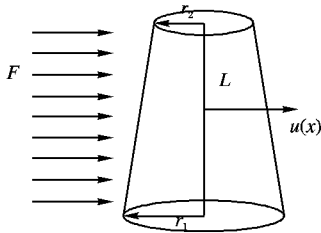


图5 Euler-Bernoulli 梁模型

表1 不同  $\alpha$  值下  $c_2, c_4$  系数及最大误差

$\alpha$	$c_2$	$c_4$	$\Delta x_{\max}$
0.05	0.221 875	0.754 029	0.046 9
0.10	0.332 600	0.398 924	0.002 4
0.20	0.374 671	0.129 428	0.008 1
0.30	0.364 816	0.245 770	0.006 0

### 2.3 长度修正

Euler-Bernoulli 梁对于小的形变有很好的效果,但是因为挠度只应用于截面方向,对于较大的偏移会表现出明显的拉伸。为了修正变换后的拉伸效果,可以对顶点位置沿着梁的切线方向进行修正。令  $s(x) = \sqrt{1 + u'^2(x)}$ ,原位置和偏移位置的差值  $d(x)$  可表示为:

$$d(x) = \frac{u(x)}{u'(x)}(s(x) - 1) \quad (15)$$

参见图6,从原位置  $p_0 = (x, y)^T$  开始,未修正的点为  $p_u = (x, y + u(x))^T$ 。最终的点  $p$  可以表示为  $p_0$  沿着切线方移动  $d(x)$  后的结果。

$$\begin{cases} p_x = x - \frac{d(x)}{s(x)} \\ p_y = y + \frac{u(x)}{s(x)} \end{cases} \quad (16)$$

对于任意一个顶点的形变计算,需要将顶点变换到枝干本地坐标系  $(r, s, t)$ ,其中  $t$  为沿着梁的轴线方向。

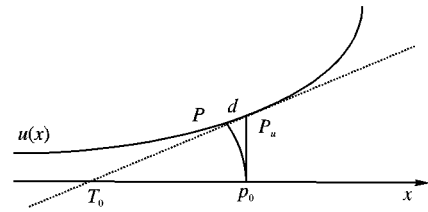


图6 长度修正

令  $\bar{x}$  为沿着  $t$  方向的坐标值,则  $x = \bar{x}/L, x \in [0, 1]$ 。顶点在本地坐标系下用偏移量可表示为:

$$\begin{aligned} p &= p_0 + o(x); \\ o(x) &= \begin{pmatrix} -d_r/S_r(x) - d_s(x)/S_s(x) \\ u_r(x)/s_r(x) \\ u_s(x)/s_s(x) \end{pmatrix} \end{aligned} \quad (17)$$

### 3 运动合成

虚拟场景中的风并不能被直接观测到,人们意识到风的存在是由于枝干以及树叶的摆动。由于枝干受风力作用是一个非常复杂的动力学过程,实时计算枝干和风场的相互作用必定会造成大量的运算负担。因此,本文采用随机的方式对枝干受风力作用的运动进行建模。

#### 3.1 风力模拟

由于外力和树的物理属性以及最终的枝干运动有着很强的关系<sup>[4,12]</sup>,使采用运动合成的方法来替求解动力学方程成为可能。通过选取和叠加不同频带的周期函数可以使最终运动有很好的视觉效果,本文使用如下函数来近似表示枝干受湍流风下的阻尼振动:

$$f(t) = 0.5 \cos(\pi t)^2 \cdot \cos(3\pi t) \cdot \cos(5\pi t) + 0.02 \sin(25\pi t) \quad (18)$$

对于不同的枝干,所受风力的大小表现为其振幅的变化,依据经验函数<sup>[13]</sup>,对不同长度的枝干,使用了不同的振幅:

$$A = L^{-0.59} \quad (19)$$

其中:  $A$  为振幅,  $L$  为枝干的长度。

同时,在预处理阶段,为每个枝干赋予了随机的相位,以保证枝干的运动不显得重复。

#### 3.2 应用梁弯曲和运动

以上形变及运动的计算都是在枝干本地坐标系下完成的。由于当前枝干的位置与其父枝干的形变结果有关,所以为了得到当前枝干中顶点的位置,首先需要计算其父枝干的偏移。

各个顶点已知其所有父层级别枝干的索引,因此可通过该索引来查找其所有父枝干的坐标转换基向量、枝干长度等信息,在 GPU 中计算并得到其偏移后的世界坐标的位置,最终叠加得到当前枝干的偏移。

### 4 实验结果及分析

本文在 PC 机上实现了树场景的实时绘制。实验的软件环境为 Windows XP 系统、VS2010、OpenGL3.3。硬件环境为 Pentium4 2.8 GHz、DDR2 1 GB、GeForce 310。分辨率为  $1280 \times 720$ 。采用了两种树木模型进行了实验,两种模型的枝干及叶片数据由表2所示。在场景完全渲染和无光照两种条件下进行了实验。无光照条件下,片元着色器不进行运算,只

输出顶点的位置信息。同时,使用 OpenGL 的 GL\_TIMESTAMP API,获得了每部分渲染在 GPU 内部耗费的精确时间,并且统计了程序的帧率(单位:帧/s)。最终场景还包含了简单地形以及天空盒子。图 7 为实验结果的部分截图。

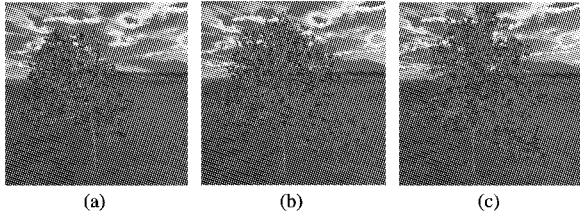


图 7 动态树场景的实验结果

表 2 不同模型的枝干及叶片数量

模型类型	枝干数	叶片数
模型 1	496	3 412
模型 2	1 084	2 199

从表 3 可以看出,系统的主要时间耗费在于叶片的动态渲染,这是因为叶片一般处于层次形变计算的最底层,为了获得叶片的位置信息需要计算上层枝干的所有信息。对于叶片的光照处理,由于使用了预计算的半透明贴图 and 法线贴图,在渲染的时候并没增加过多的负担。

表 3 不同实验条件下的统计数据

实验条件	枝干渲染 时间/ms	叶片渲染 时间/ms	帧率 (帧/s)
模型 1(有光照)	3.81	24.54	32
模型 2(有光照)	7.63	15.67	40
模型 1(无光照)	3.41	21.08	34
模型 2(无光照)	5.24	13.11	41

另外,由于在 CPU 端需要处理输入、输出以及和 GPU 同步等问题,这使得在部分实验中 GPU 的运算并非系统的瓶颈,表现为最终帧率和 GPU 运算时间不成直接反比关系。

## 5 结语

本文通过应用 Cook-Torrance 光照模型和半透明贴图增强了树木的真实感,拟合得到的挠度公式结合长度修正使得形变的计算得以快速进行,层次化枝干贴图索引的方式充分发挥了 GPU 的并行性。实验结果表明,该方法可以较为真实

地渲染出树的动态效果。

需要指出的是,该方法不适用于风速的急剧变化的场景。对于不同类型的风场的建模还有待进一步的研究和改进。同时,使用新的编程接口、几何着色器等技术充分发挥 GPU 的性能也是今后工作的一个重点。

## 参考文献:

- [1] BEHRENDT S, COLDITZ C, FRANZKE O, *et al.* Realistic real-time rendering of landscapes using billboard clouds[J]. Computer Graphics Forum, 2005, 24(3): 507-516.
- [2] SOUSA T. Vegetation procedural animation and shading in crysis[M]. Upper Saddle River: Addison-Wesley, 2007: 373-407.
- [3] KHARLAMOV A, CANTLAY I, STEPANENKO Y. Next-generation speedtree rendering[M]. Upper Saddle River: Addison-Wesley, 2007: 69-92.
- [4] ZIOMA R. GPU-generated procedural wind animations for trees[M]. Upper Saddle River: Addison-Wesley, 2007: 231-240.
- [5] 李占利, 千叶则茂. 树在风中摆动的计算机实时模拟方法研究[J]. 系统仿真学报, 2008, 20(8): 2085-2090.
- [6] HU S J, CHIBA N, HE D. Realistic animation of interactive trees[J]. The Visual Computer, 2012, 28(6/8): 859-868.
- [7] YANG M, HUANG M C, WU E H. Physically-based tree animation and leaf deformation using CUDA in real-time[M]. Berlin: Springer-Verlag, 2011: 27-39.
- [8] HABEL R, KUSTERNIG A, WIMMER M. Physically based real-time translucency for leaves[C]// Proceedings of the 18th Eurographics Symposium on Rendering. Aire-la-Ville, Switzerland: Eurographics Association, 2007: 253-263.
- [9] COOK R L, TORRANCE K E. A reflectance model for computer graphics[J]. ACM Transactions on Graphics, 1982, 1(1): 7-24.
- [10] HABEL R, KUSTERNIG A, WIMMER M. Physically guided animation of trees[J]. Computer Graphics Forum, 2009, 28(2): 523-532.
- [11] 王春华, 李留青. 基于 GPU 的风中树木物理动画[J]. 计算机应用研究, 2010, 27(10): 3988-3990.
- [12] OTA S, FUJIMOTO T, TAMURA M, MURAOKA K, *et al.* 1/f<sup>β</sup> noise-based real-time animation of trees swaying in wind fields[C]// Proceedings of Computer Graphics International 2013. Washington, DC: IEEE Computer Society, 2003: 52-59.
- [13] CODER K D. Sway frequency in tree stems[EB/OL]. [2012-10-20]. <http://warnell.forestry.uga.edu/service/library/for00-024/for00-024.pdf>.

(上接第 1676 页)

- [3] SUBRAHMANYAM M, MAHESHWARI R P, BALASUBRAMANIAN R. Local maximum edge binary patterns: A new descriptor for image retrieval and object tracking[J]. Signal Processing, 2012, 92(6): 1467-1479.
- [4] 余胜, 曾接贤, 谢莉. 基于多特征整合的图像检索算法[J]. 计算机工程, 2012, 38(24): 216-219.
- [5] HU R X, JIA W, ZHAO Y, *et al.* Perceptually motivated morphological strategies for shape retrieval[J]. Pattern Recognition, 2012, 45(9): 3222-3230.
- [6] 王向阳, 陈景伟, 于永健. 一种基于彩色边缘综合特征的图像检索算法[J]. 模式识别与人工智能, 2010, 23(4): 216-221.
- [7] WAHAB M H A, HSIEH T M. Image retrieval based on color and texture features[C]// Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery. Washington, DC: IEEE Computer Society, 2012: 1816-1819.
- [8] VIMINA E R, POULOSE JACOB K. Image retrieval using colour and texture features of regions of interest[C]// 2012 International Conference on Information Retrieval & Knowledge Management. Washington, DC: IEEE Computer Society, 2012: 240-243.
- [9] LIN C H, CHEN R T, CHAN Y K. A smart content-based image retrieval system based on color and texture feature[J]. Image and Vision Computing, 2009, 27(6): 658-665.
- [10] LIU G H, ZHANG L, HOU Y K. Image retrieval based on multi-texton histogram[J]. Pattern Recognition, 2010, 43(7): 2380-2389.
- [11] 邹彬, 潘志斌, 胡森. 基于局部投影与块 LBP 特征的图像检索[J]. 中国图象图形学报, 2012, 17(6): 671-677.
- [12] 王向阳, 李东明, 杨红颖. 基于 Zernike 色度分布矩的彩色图像检索算法[J]. 模式识别与人工智能, 2012, 25(2): 313-319.