基于 GPU 的风中树木物理动画 *

王春华, 李留青
(黄淮学院 计算机科学技术系, 河南 驻马店 463000)

摘　要: 采用非均匀梁作为树枝物理模型,分析了梁的弯曲效应,推导出挠度曲线表达式,并对其运用多项式最小二乘法拟合,将拟合后的简洁表达式预计算于纹理中,有效降低了实时计算量。为平衡 CPU 与 GPU 的负载,设计了分层次的树木结构模型,将力学分析计算转移到 GPU 上进行,同时避免了将树枝分段计算导致段间断裂现象及分段过大引起模拟效果过于粗糙的缺点。实验表明,该方法能完全实时、真实地模拟出风中树木动画的物理效果,解决了目前树木物理动画速度太慢不能实时模拟的不足。

关键词: GPU; 风; 树木; 物理动画
中图分类号: TP391　　**文献标志码**: A　　**文章编号**: 1001-3695(2010)10-3988-03
doi:10.3969/j.issn.1001-3695.2010.10.106

Physical animation of trees in wind based on GPU

WANG Chun-hua, LI Liu-qing
(*Dept. of Computer Science, Huanghuai University, Zhumadian Henan* 463000, *China*)

Abstract: This paper introduced a new real-time tree animation method that was both significantly faster and more physically plausible than previous methods. Taking physical model of branches as linearly tapered circular beam, analyzed its bending effect and derived the expression of its deflection curve. To enhance computational efficiency, performed a least square fit to the deflection curve equation and then pre-computed its value to 2D-texture which was accessed by GPU run-time. To balance the load on CPU and GPU, this paper designed a hierarchical tree structure model that transfered all the mechanical analysis and calculation to GPU, meanwhile, avoided to segment a branch to model its deformation behavior. Experiments show that the presented method can be completely real-time, realistically simulate the tree in wind, solving the problem of current method in literature that cannot simulate the physical real-time animation of tree in wind.

Key words: GPU; wind; tree; physical animation

树木是自然场景中不可或缺的重要组成部分,而树木在风中的动画是影响树木真实感的决定性因素之一。实时、逼真地模拟树木对风的响应是一项具有挑战性的课题,主要原因在于树木几何结构的高度复杂性。为了真实模拟树木对风的响应,以下三方面内容需要明确:

a)风力模型,用于描述风的行为特征;

b)力学模型,用于分析相互连接的树枝和树叶对风力的响应;

c)树木结构模型,用于描述树木几何元素等级组织关系。

目前,树木在风中的动态模拟主要有两类方法:过程式方法[1~3]和基于物理的方法[4~8]。基于过程式的方法没有考虑树木的物理特性,采用经验公式计算树的位移,效率较高,但需反复试验才能得出适宜的公式,不具有通用性。基于物理的方法通过求解力学方程得到树的形变,能比较真实地模拟树木在风中的运动,但是计算开销太大,不适于实时计算场合。在这两类模拟方法中,树木结构模型盛行采用蒙皮骨骼动画(skinned skeletal animation),将分枝用分段的刚性杆件表示,该杆件不能变形只能弯曲[1,4,6]。该结构模型具有三个缺点:a)将所有计算负载置于 CPU 上,因为每个分枝的骨骼矩阵(bone matrix)必须在每帧中重新计算;b)通过增加更多的连接点(joint)来提升模拟枝条弯曲的精细程度,这将导致更大的计算量;c)相邻分段间的弯曲程度不同时会产生裂口现象。以上几点原因导致基于连接点的方法不适合用于高质量的实时树木动画。

本文提出一种实时模拟树木与风场交互的方法,所有的变形计算都基于 GPU。树木结构模型采用分级层次表示,适合在 GPU 中计算不能访问上次运算结果的特点。为了简化计算、提高计算效率,在不影响真实感的前提下,本文弱化了树枝间的联系,采用 Euler-Bernoulli 梁理论分析树枝的受力与挠度的变化关系。运用线性最小二乘法拟合挠度曲线表达式为简洁形式,并将简化后的表达式预计算于纹理中,实时计算时通过 GPU 顶点纹理采样访问,大幅度地降低了计算开销,是本文实时模拟风中树木运动的关键所在。

1　基于 GPU 的树木结构模型

为提高计算效率以达到实时模拟目的,将树枝的所有变形计算移植到顶点着色器(vertex shader)上进行,平衡 CPU 与 GPU 的工作量,充分发挥 GPU 高效计算的优势。借签文献[3]中的部分思想,树木结构模型采用分层次结构,不同之处在于,本文考虑了顶点在本枝中的位置索引,而文献[3]只考虑了顶点在它所有父枝中的索引。首先,将树木结构分层描述,如图 1 所示,借用二叉树结构中各层次的概念描述,兄弟

收稿日期: 2010-02-04; **修回日期**: 2010-04-21　　**基金项目**: 2009 和 2010 年河南省科技发展计划项目(092102210376,102102210414)
作者简介:王春华(1980-),女,博士,主要研究方向为计算机图形学、自然景物建模(sunrisefe@126.com);李留青(1981-),女,讲师,硕士,主要研究方向为计算机图形学、电子商务.

分枝为具有相同父分枝的各树枝,具有子树枝的树枝称为子树枝的父分枝。其次,为了显式计算所有父分枝的受力变形且无须在运行期间传递任何信息,需要将树木结构(如层级信息)的相关信息暴露给每个顶点。为了达到这个目标,每个树枝顶点应该知道它所在层级的索引。根据树木本身的特点,设计出的树木结构模型如下:

a)对于具有相同父分枝的兄弟树枝而言,它们具有相同的父分枝索引信息;

b)对于同一枝条上的各个顶点而言,它们除了具有相同的父分枝索引信息外,还存储了它们在相同枝条上的位置索引。

不同的兄弟树枝在相同父树枝上的位置关系,采用规范化的 $x \in [0,1]$ 表示,具体为

$$x = \frac{c - p_0}{p_1 - p_0} \tag{1}$$

其中:c 为子树枝在父树枝上的起始位置,p_0 和 p_1 分别为父枝的起点位置和终点位置,位置相减采用欧拉距离计算。每条树枝的索引值 $\in [0,1]$,其中:0 表示父树枝起点,1 表示父树枝终点。假设树木具有三层结构,如图 2 所示,那么每个 0 级树干上各顶点的索引信息只有一个,1 级树枝上各顶点的索引信息有两个(在 1 级树枝上的位置索引和在 0 级树主干上的位置索引),2 级树枝上每个顶点的索引信息有三个(在 2 级树枝、1 级树枝以及 0 级树主干上的位置索引)。

2　风力模型

树木在风中的动画主要源于风的变幻,为模拟出真实的动画效果,反映风力自然规律的模型必不可少。根据大气研究方面的知识,瞬时风速可表示为平均风速和脉动风速之和[7]:

$$v = \bar{v} + v_{dp} \tag{2}$$

由于脉动分量 v_{dp} 相对于平均风速 $\bar{v}$ 成对称分布,且大体上符合正态分布,故常将它近似为高斯过程。

顺着风向,整个风力由拖曳力和升力两个分量构成,根据风速风压关系公式拖曳力和升力分别为:

$$F_d = \frac{1}{2} C_D \rho_\alpha v^2 S \tag{3}$$

$$F_t = \frac{1}{2} C_L \rho_\alpha v^2 S \tag{4}$$

其中:C_D 和 C_L 分别是阻力系数和升力系数;S 表示迎风面积;ρ_α 为空气密度。

根据文献[9]提供的数据,取 C_D 和 C_L 系数分别为 0.013、0.75,空气密度 $\rho_\alpha = 2.37 \times 10^{-3}\ \mathrm{g/m^3}$。

3　力学模型

树枝的长度远大于直径,且树木在风中的舞动效果主要体现为树枝的弯曲,枝条的剪切变形影响可以忽略,故本文采用欧拉梁来模拟树枝。梁的几何特性和物理特性的模型决定了树枝在风中舞动的真实感。在真实动画中,经常采用弹性圆柱(均匀梁)表示梁的几何形体,并用均匀横向力计算梁的变形[9]。但是,均匀梁并不能很好地逼近树枝,本文的模拟方法采用了一种更接近于真实树枝的梁模型,考虑了树枝的基本物理属性。

3.1　非均匀梁受力弯曲变形分析

将树枝视为图 3 所示具有线性锥度比的梁,长度为 L,半径 s_1 和 s_2 分别表示固定端与自由端的半径。

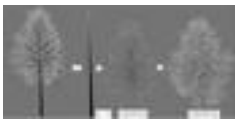

图1　树木的分级图

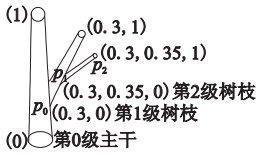

图2　树枝上各顶点的位置索引

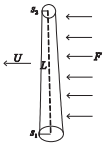

图3　用于计算挠度的悬臂梁

树枝的挠度由欧拉—柏努利(Euler-Bernoulli)微分方程描述:

$$\frac{\partial^2}{\partial x^2}\left(EI(x) \frac{\partial^2 u(x)}{\partial x^2} \right) = F \tag{5}$$

其中:$u(x)$ 挠曲线表示位置 x 处的挠度;I 是截面惯性矩(与所受载荷方向有关);F 为分布荷载;E 是弹性模量。

因为树枝一端固定在父枝(或树主干)上,称该端为固定端,所以该端的挠度和斜率都为 0,即它的边界条件为

$$u\Big|_{x=0} = 0 \text{ 且 } \frac{\partial u}{\partial x}\Big|_{x=0} = 0 \tag{6}$$

树枝自由端的剪力和弯矩为 0,即边界条件为

$$\frac{\partial^2 u}{\partial x^2}\Big|_{x=L} = 0 \text{ 且 } \frac{\partial^3 u}{\partial x^3}\Big|_{x=L} = 0 \tag{7}$$

为了简化计算并应用于本文设计的基于 GPU 的树结构模型中,将梁模型规范化为单位长度,并用锥度比 α 表示树枝两端的半径比。

$$r_{1,2} = \frac{s_{1,2}}{L} \quad \alpha = \frac{r_2}{r_1} \tag{8}$$

梁的圆形截面的截面惯性矩为

$$I = \pi r^4 / 4 \tag{9}$$

是半径 r 在梁的轴方向线性变化。在此,采用具有线性锥度比的梁建模树枝的重要性通过半径 r 的四次方得以充分体现,当半径变化时挠度也会变化。由式(8)和(9)可推导出沿着梁轴方向的截面惯性矩可表示为

$$I(x) = \frac{\pi r_1^4 ((\alpha - 1)x + 1)^4}{2} \tag{10}$$

在给定的边界条件式(6)(7)和可变的截面惯性矩(10)时,欧拉—柏努利微分方程(5)的解析解为

$$\begin{aligned} u(x) = [& F((x(a-1)(2x(a-1)(3+(a-1)a) + \\ & 3(4(a-2)a))) - 6(1+x(a-1))^2 \ln(1+x(a-1)))] \\ & / [r_1^4 LE(3\pi(1+x(\alpha-1))^2(a-1)^2)] \end{aligned} \tag{11}$$

从式(11)可知,力 F 与挠度值成正比(即挠度 $u(x)$ 的幅值与力 F 成正比),与树枝固定端半径 r_1^{-4} 成正比。

为了提高在顶点着色器中计算 $u(x)$ 的效率,采用多项式拟合最小二乘法简化挠曲线等式(11)为多项式:

$$u(x) = \frac{F}{r_1^4 LE}(c_3 x^3 + c_2 x^2 + c_1 x + c_0) \tag{12}$$

其中:$c_0 \sim c_3$ 是与 α 有关的系数,表 1 为不同锥度比时 $c_0 \sim c_3$ 的取值及误差绝对值的最大值 $\Delta x_{\max}$。通过多项式拟合,$u(x)$ 的计算量从 58 次降为 19 次,计算量为原来的 32.8%。顶点纹理采样是随着图形处理器的发展而提供的一种顶点采样功能,在 vs_3_0 以上版本得到支持。将拟合后的多项式部分 $c_3x^3 + c_2x^2 + c_1x + c_0$ 视为自变量为 α 和 x 的二元函数,α 和 x 分别对应 GPU 纹理的 u、v 坐标,则可将多项式预计算存储于纹理中,实时计算时通过顶点纹理采样访问预计算信息。GPU 实施顶点纹理采样只需一次操作,所以挠度曲线 $u(x)$ 的计算从 58 次减少为 8 次,为原计算量的 13.8%。

表 1　多项式拟合系数及误差

α	c_3	c_2	c_1	c_0	Δx_{max}
0.01	4.643907	-4.534121	1.445770	-0.088322	0.095930
0.04	2.254374	-1.749001	0.582371	-0.034076	0.028088
0.07	1.346111	-0.767442	0.293268	-0.016858	0.010857
0.10	0.930822	-0.400027	0.217248	-0.017686	0.003100
0.13	0.559550	-0.002033	0.085087	-0.006290	0.001026
0.14	0.47192	0.086343	0.05658	-0.0038766	0.001034
0.20	0.147710	0.379100	-0.033284	0.003490	0.001568
0.30	-0.053461	0.493760	-0.060946	0.005368	0.001209

3.2　三维空间弯曲变形计算

上节推导了二维空间梁挠度计算公式,为了将其应用到任意树枝的顶点上,需要在树枝局部坐标系中表示挠度。假设每个树枝的顶点由树枝本地坐标系(t,r,s)表示,t 为沿梁轴方向,如图 4 所示,左图为变换前树枝的局部坐标系(与世界坐标系相同),右图为变换后的局部坐标系。将风力投影到 r 轴和 s 轴上分别计算其挠度 u_r、u_s,最终树枝的变形位移为

$$p' = p_0 + r \times u_r + s \times u_s \tag{13}$$

其中:p_0 为变形前的位置,p'是变形后的位置。注意,由于每层树枝上各顶点的变形除了本级树枝的受力变形外,还与它的所有父分枝的变形有关,即子枝的变形是当级树枝变形叠加上所有父枝的变形。以图 2 中的点 p_2 为例,顶点 p_2 受力变形后的最终位置表示为 p',如式(14)所示。其中:p_2 为变形前的初始位置;p_2'表示顶点 p_2 来第 2 级分枝上的变形偏移;p_1'表示顶点 p_1 在第 1 级分枝上的变形偏移;p_0'表示 p_0 在树主干上的变形偏移。

$$\begin{cases} p' = p_2 + p_2' + p_1' + p_0' \\ p_2' = r_2 \times u_{r2} + s_2 \times u_{s2} \\ p_1' = r_1 \times u_{r1} + s_1 \times u_{s1} \\ p_0' = r_0 \times u_{r0} + s_0 \times u_{s0} \end{cases} \tag{14}$$

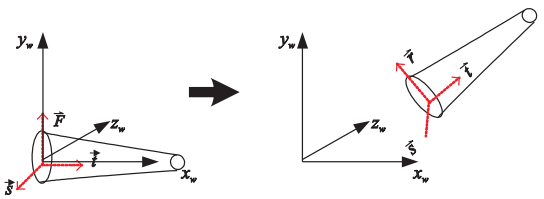

图4　树枝的本地坐标系

4　实验及分析

4.1　简要实现

树枝的每个顶点及每个树枝需要的信息参见表 2,为了在 GPU 上快速访问每个顶点所在树枝的信息,将树枝信息存储在纹理中,在顶点着色器上直接进行顶点纹理采样。

表 2　顶点及树枝信息表

每个顶点	位置 pos、在各级的索引信息 pos_index[ ],分枝索引 branch_index
每条树枝	分枝索引 branch_index,分枝长度 length,分枝锥度比 α,局部坐标系($\vec{x}_l,\vec{y}_l,\vec{z}_l$),分枝层次 level,父分枝 parents[ ],杨氏模量 E

GPU 顶点着色器上的算法实现描述如下:

全局变量

float branch_index, level, parent, E,α,length;

//各参数含义参见表 2

vector3: x,y,z;//分枝所在的局部坐标系

顶点着色器

1)得到该顶点所在分枝的索引 branch_index;

2)顶点采样查询顶点所在分枝的层次 level;

3)while(level≥0.1)

{

(1)顶点采样查询该分枝的锥度比 α、杨氏模量 E、分枝长度 L、分枝的局部坐标系($\vec{x}_l,\vec{y}_l,\vec{z}_l$)、父分枝 parent;

(2)根据 α 值查询多项式拟合后的系数 c_2 和 c_4;

(3)由式(12)计算顶点在该级分枝上的弯曲后的位置 position;

(4) - - level;

}

4.2　实验分析

在显卡为 NVIDIA GeForce 9800 GTX + 、CPU 为 DualCore Intel Core 2 Duo E7400、内存大小为 3327 MB 平台上,采用 DirectX9 图形开发包进行编程实验。图 5 和 6 分别是一棵树在不同风力和不同风向下实验的结果 帧率统计如表 3 所示。

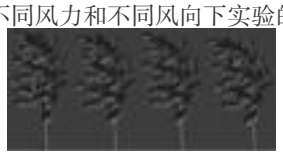

图5　在不同风力作用下的示意图

图6　不同风向下的模拟示意图

表 3　帧率对比表

顶点数	三角面片数	帧率
22 782	25 480	1 119
23 816	26 420	1 117
34 922	38 080	1 104
56 260	59 980	1 102

从图 5、6 可知,本文设计的基于 GPU 的树木动画模拟方法有效,模拟结果逼真;从表 3 可知,本文的树木动画模拟方法高效(帧率达 1 119 fps)且随顶点数目增加帧率影响不大,从本质上提高了绘制速度,而以往文献的模拟速度仅为几帧,难以达到实时(通常为 30 fps)。

5　结束语

本文设计了基于 GPU 的分层次树木结构模型,采用非均匀梁作为树枝模型,分析并推导了树枝变形的挠度曲线表达式,运用线性最小二乘法拟合挠度曲线并将其预计算到顶点纹理中,在 GPU 上实时计算树枝各顶点由于风力作用引起的位移时顶点纹理采样预计算结果,大幅度地降低了计算量。实验结果表明,本文提出的树木在风中运动的物理方法高效、模拟结果逼真,能广泛应用于虚拟场景模拟中。

参考文献:

[1] SHIN O, FUJIMOTO T, TAMURA M, *et al*. 1/*f* beta noise-based real-time animation of trees swaying in wind fields[C]//Proc of International Computer Graphics Conference. 2003: 52-59.

[2] WESSLÉN D, SEIPEL S. Real-time visualization of animated trees [J]. The Visual Computer, 2005, 21(6): 397-405.

[3] ZIOMA R. GPU-generated procedural wind animations for trees: NGUYEN GPU Gems 3[M]. [S. l.]: Addison Wesley, 2007.

[4] AKAGI Y, KITAJIMA K. Computer animation of swaying trees based on physical simulation[J]. Computers and Graphics, 2006, 30(4): 529-539.

[5] ZHANG Long, SONG Cheng-fang, TAN Qi-feng, *et al*. Quasi-physical simulation of large-scale dynamic forest scenes[C]//Proc of International Computer Graphics Conference. 2006: 735-742.

[6] 冯金辉, 严涛, 陈彦云, 等. 树木的整体性运动及树内部风场的研究[J]. 软件学报, 2000, 11(3): 363-367.

[7] 柳有权, 王文成, 吴恩华. 快速真实地生成树的自然摇曳[J]. 计算机学报, 2005, 28(7): 1185-1192.

[8] 胡潇毅, 李华锋, 郭乙木, 等. 基于动力学有限元的树在风中摇曳动画[J]. 计算机辅助设计与图形学学报, 2007, 19(9): 1166-1171.

[9] BOURG D M. Physics for game developers[M]. Beijing: Publishing House of Electronics Industry, 2002: 135-137.