

草地场景碾压效果模拟

邱航¹ 陈雷霆¹ 蔡洪斌¹ Jim X. Chen²

(电子科技大学计算机科学与工程学院 成都 611731)¹ (美国乔治梅森大学计算机系 费尔费克斯 22030)²

摘要 草体作为自然场景的一种重要元素,数量众多、覆盖范围广,很难实现实时模拟。针对已有方法存在的问题,提出了一种草体碾压效果模拟方法。利用混合式表达方式构建出大规模静态草地场景,采用基于GPU的实时碰撞检测算法和受力向量传播机制计算视点近处草叶的受力和形变,通过调整布告板斜率模拟中远距离草丛变形。为消除混合式表达方法带来的跳动现象,实现了一种层间动态过渡策略。实验结果表明,该方法能模拟出逼真的草体碾压效果,同时有效弥补了传统算法的缺陷。

关键词 动态模拟,草地场景,布告板,细节层次

中图分类号 TP391 **文献标识码** A

Simulation of Treading on Grassland

QIU Hang¹ CHEN Lei-ting¹ CAI Hong-bin¹ Jim X. Chen²

(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)¹

(Department of Computer, George Mason University, Fairfax 22030, USA)²

Abstract Grass is one of the nature elements that is difficult to simulate in real-time due to its large number and wide covering range. In view of the deficiencies of existing methods, a method for simulation of treading on grass was proposed. Constructed large-scale static grassland based on hybrid representation. Computed the deformation of grass close to the viewer by using GPU-based real time collision detection algorithm and force vector diffusion mechanism. Simulated the deformation of tussock by adjust the slope of billboard. In order to avoid popping between different levels, a transition scheme was implemented. Experiments show that this method not only can realistically simulate the effect of treading on grass, but also can overcome the deficiencies of the traditional methods.

Keywords Dynamic simulation, Grassland scene, Billboard, Level of detail

1 引言

多年来,草地场景的真实感模拟一直是计算机图形学领域的研究热点^[1-5],它在视景仿真、计算机辅助园林设计、娱乐游戏等领域有着广阔的应用前景。近些年,随着计算机硬件性能和图形显卡可编程能力的不断提升,真实自然条件下的草地场景动态模拟受到了广泛的关注。

在现实世界中,草体具有柔性的结构,在与运动物体的交互中,草体会发生变形。对这一现象的真实感模拟,有助于提高虚拟场景的沉浸感。然而,由于碾压物运动规律的不确定性,再加上场景中庞大的草体数量,要快速、准确地计算草叶实体在碾压物作用下的形变,是一个难题。

目前,国内外在草体碾压效果模拟方面的研究工作较少。2003年,Guerraz等^[6]首次提出了草体上的碾压、踩踏效果模拟方法,但该方法仅根据一个受影响区域来判断草叶是否与碾压物发生碰撞,没有实际的碰撞检测计算,难以真实地反映碾压物对草叶的作用。2007年,牛立新^[7]采用粒子来表现草

叶的形变,通过在粒子和碾压物体间进行碰撞检测,利用胡克定律和动量定理计算出每个粒子新的位置,模拟出草叶被碾压后的形状。但该方法计算开销大,此外不支持任意外形的碾压物。2009年,Zhao等^[8]采用基于GPU的碰撞检测方法,将碾压物包围盒的几何信息和运动参数传递到GPU中,通过GPU完成草叶与碾压物的碰撞检测,根据不同的碰撞情况,对草叶采取不同的变形方法。但由于采用基于包围盒的碰撞检测,故碰撞精度不高。此外,该方法也没有考虑草叶与草叶之间的相互作用,与现实世界的情况并不相符。

针对以上问题,本文提出了一种实现草体碾压效果模拟的方法,实现了任意形状物体与草叶实体间的可控精度碾压效果模拟。

2 草地场景的建模

2.1 草地场景混合式表达方法

与树木等其他植被相比,草体的几何结构相对简单。然而在大规模草地场景中,草叶实体种类繁多、形状各异,很难

到稿日期:2010-07-23 返修日期:2010-12-17 本文受国家863项目(2007AA010407),航空基金项目(20080580005),总装“十一五”预研基金项目资助。

邱航(1978—),男,博士生,讲师,CCF会员,主要研究方向为计算机图形学,E-mail: qiuhan@uestc.edu.cn;陈雷霆(1966—),男,博士,教授,博士生导师,CCF高级会员,主要研究方向为虚拟现实技术、计算机图形学;蔡洪斌(1966—),男,博士,教授,CCF高级会员,主要研究方向为计算机图形学;Jim X. Chen(1962—),男,博士,教授,博士生导师,主要研究方向为虚拟现实技术、计算机图形学。

通过统一的场景建模方法进行绘制。本文采用混合式表达方法来构建静态草地场景,通过不同精度的模型表达手段构造出多层次局部草地场景,根据视点的变化选择合适的方法来展现草叶的视觉外观。

与文献[1,6,8]类似,我们根据草叶实体距离视点的远近,将场景划分为近距离、中距离和远距离3个层次。在每个层次中采用不同的草体表达方法,即场景中的草体对应了3种不同的层次细节。

如图1所示, P 点为视点在场景中的位置,三角形区域为视点的视野范围区域。根据离视点的远近,将视野范围划分为3个区域,不同区域内的草体采用不同的表达方法:近处利用几何的方式展现草体细节,中间区域采用布告板表达草丛,远处直接通过2D纹理代表草场远景。这种混合式表达方法不仅可以有效减少场景中多边形的数量,而且符合人们观察景物的经验。

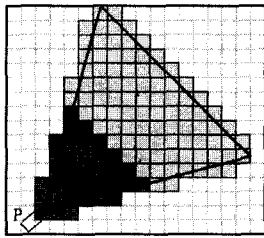


图1 草地场景 LOD 表达

为了对整个场景实施有效的管理和加速绘制,本文采取了分块管理的策略,如图2所示。

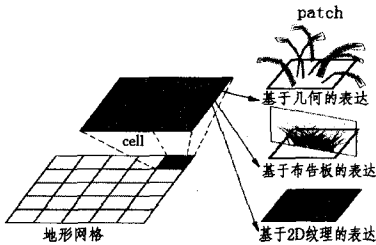


图2 分块管理策略

首先将整个场景地形均匀划分为若干大小相同的单元,每个单元称为 cell,它是绘制管理的基本单位,即各 cell 独立控制自身的绘制过程。同时,cell 也是进行基于视锥体剪裁的基本单位,在绘制前将视野以外的场景以 cell 为单位进行剔除,从而减少计算量。进一步将每个 cell 均匀划分为若干称为 patch 的单元小格,patch 是草体表达的基本单位,即各 patch 独立控制本单元中草叶实体的表达方法以及数量。按照上述“cell-patch”两级分块策略划分整个场景,最后按照一定的随机策略将相应草体模型植入 patch 中,实现对整个场景中对象的管理。

2.2 LOD 过渡策略

由于场景中存在3种层次的草体表达方法,各层间具有非统一性,因此需要在层与层的交接处进行必要的处理,以避免视点移动时出现视觉上的突变。我们在相邻层次间设置过渡区域,采用相邻两种表达方法共存的方式进行绘制,通过权重函数来控制不同表达方法在过渡区域中所占的比重。

在三角形与布告板表达方法间的过渡区域,布告板权值 $K_s(d)$ 与三角形权值 $K_g(d)$ 分别为

$$K_s(d) = (d - d_{g1}) / (d_{g2} - d_{g1}) \quad (1)$$

$$K_g(d) = 1 - K_s(d) \quad (2)$$

式中, d 为视点 to 草叶所在 patch 的距离; d_{g1} , d_{g2} 分别为过渡区域范围相对于视点的最近及最远距离; $K_s(d)$ 代表顶点的透明度, $K_g(d)$ 对应该 patch 中草叶的根数 K :

$$K = \lfloor n \cdot K_g(d) \rfloor \quad (3)$$

n 为采用多边形表达时,一个 patch 中允许植入的草叶最大数量。

同理,在布告板与2D水平纹理表达方法间的过渡区域,水平纹理权值 $K_h(d)$ 与布告板权值分别为

$$K_h(d) = (d - d_{s1}) / (d_{s2} - d_{s1}) \quad (4)$$

$$K_s(d) = 1 - K_h(d) \quad (5)$$

式中, d_{s1} , d_{s2} 分别为过渡区域范围相对于视点的最近及最远距离。布告板权值和水平纹理权值均代表了顶点的透明度。

3 草体碾压效果模拟

3.1 框架描述

我们根据草叶距离视点的远近以及其与碾压物的接触关系,将草叶实体分为直接形变草叶、连带形变草叶以及 billboard 草丛3种类型并分别进行处理。直接形变草叶与连带形变草叶均为近距离基于多边形表达的草叶实体,其中直接形变草叶与碾压物发生碰撞接触时,我们采用基于 GPU 的碰撞检测算法确定直接形变草叶位置并完成形变计算。连带形变草叶的形变是由周围草体的相互作用产生,为了简化计算,不进行草叶间的碰撞检测,而是采用受力向量传播机制获取其受力向量并计算形变;billboard 草丛距离视点较远,无需精确的碰撞计算,根据碾压物与 billboard 的距离,通过控制 billboard 的斜率来表现草丛的变形。整个模拟过程分为离线预处理和实时处理两个阶段,如图3所示。

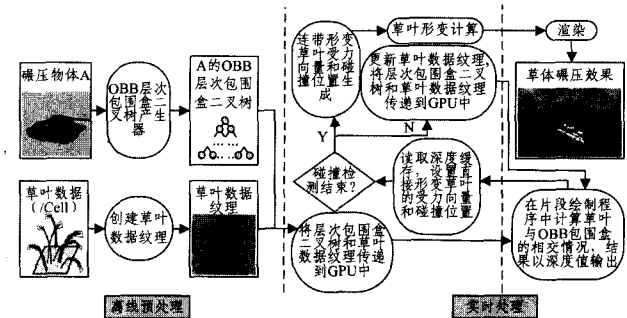


图3 草体碾压效果模拟框架

离线预处理阶段主要完成碾压物模型数据的处理、草叶数据创建等工作,为碾压物和草叶之间的实时碰撞检测做准备;实时处理阶段完成实时的碰撞检测、草叶形变计算以及场景绘制工作。下边对以上两个阶段的工作流程进行详细介绍。

(1) 离线预处理阶段:首先导入碾压物模型,创建碾压物的 OBB 层次包围盒二叉树,然后创建草叶数据,并将草叶数据以 cell 为单位按一定方式组织到纹理中。至此,所有数据已准备完成,离线预处理结束。

(2) 实时处理阶段:首先进行碾压物体和草叶间的碰撞检测,将离线预处理阶段创建的 OBB 层次包围盒树和草叶纹理数据传递到 GPU 中,利用 GPU 的并行处理特性,通过片

段绘制程序在 GPU 中完成碰撞检测主要的计算工作,得到所有被碰草叶(直接形变草叶)的信息,包括碰撞的位置、受力向量等。碰撞检测只确定了与碾压物直接交互的草叶,因此采用一种类似种群繁衍的方法传递受力向量,生成连带形变草叶的受力向量。在形变计算阶段,根据不同的草叶类型,采用相应策略计算形变。最后,完成整个场景的绘制。

3.2 基于 GPU 的实时碰撞检测

要逼真展现不同外形碾压物与几何草体的作用关系,首先需要进行碰撞检测。我们以基于流的实时碰撞检测算法^[9]为基础,充分利用图形硬件的并行架构,结合草叶特性,以 cell 为单位实现可控粒度的碰撞检测计算,其主要过程包括:

(1) 构造碾压物的 OBB 层次包围盒二叉树,同时将草叶数据以 cell 为单位组织到纹理中。

(2) 将碾压物的包围盒与每个 cell 中草叶形成的包围盒进行相交测试,裁剪掉不相交的 cell,从而大幅削减参与碰撞检测的草体数量。

(3) 对于相交的 cell,以 cell 为单位,将其包含的草叶与碾压物 OBB 层次包围盒树根节点做相交检测,结果作为深度值输出。读取深度缓存,得到发生相交的草叶的集合。如果集合为空,则该 cell 中的所有草叶都不与碾压物相交。否则,将相交的草叶数据重新组织到纹理中,递归判断草叶与下面两个子节点的相交情况,直至达到某一满足特定精度需求的树层次。这里的精度是指碰撞检测所进行到的 OBB 层次包围盒树的层次。

3.2.1 草叶数据组织

对于近距离草叶,由于在建模时采用四边形条带构成整根分节草体模型,因此将草叶的骨架线数据组织到纹理中进行碰撞检测,即纹理中每个像素的 3 个颜色通道存放骨架线上的一个顶点坐标。

一个包含 m 根草叶的 cell,设每根草叶具有 n 个分节,则需要 $n+1$ 张纹理存储草叶数据,其中第 $k(k=0,1,2,\dots,n)$ 张纹理存储 cell 中每根草叶骨架线上对应的第 k 个顶点坐标。骨架线顶点数据在纹理中的位置 (i,j) 与该草叶在 cell 中的相对位置 (i,j) 一致,纹理的长 W 和宽 H 由 cell 中草叶数量 m 决定:

$$W=H=\lceil\sqrt{m}\rceil \quad (6)$$

草叶数据与顶点纹理的对应方式如图 4 所示。

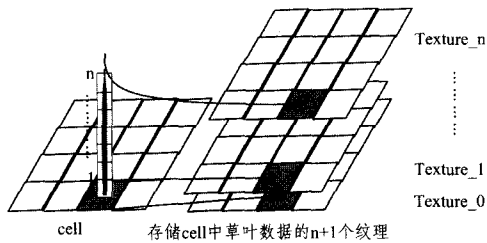


图 4 草叶数据组织

3.2.2 草叶与碾压物碰撞检测

碰撞检测由 CPU 和 GPU 协作共同完成。片段处理程序完成草叶与 OBB 包围盒的相交检测,CPU 通过深度缓存读取相交检测的结果,将发生碰撞的草叶组织成新的纹理传到 GPU 中,继续进行相交检测。

首先,将 OBB 层次包围盒树的根节点作为当前节点,进

行草叶与当前节点的 OBB 包围盒的相交检测。草叶与碾压物 OBB 包围盒求交需要两个部分的数据:

(1) 草叶骨架线 n 条边的信息(假设草叶有 n 个分节)。首先创建一个分辨率大小与草叶数据纹理相同的视口,然后采用平行投影的方式建立一个大小与视口相同的矩形。通过纹理映射,将草叶数据纹理映射到矩形的各个像素上,这样就可以得到视口中每个像素和 cell 中的每根草叶的一一映射。在片段绘制程序中,通过纹理采样获得草叶骨架线的信息。

(2) 定义碾压物 OBB 包围盒的数据。主要包括一个中心点 C 、一个确定方向和位置的基底向量为 $[\vec{u}, \vec{v}, \vec{w}]$ 的正交系以及 3 个分别表示半宽、半高和半深的长度数据 hu, hv, hw ,由片段绘制程序的参数传入。

我们首先将 OBB 层次包围盒树的根节点作为当前节点,采用一种可用于线形对象与有向有界箱的求交算法^[10],在片段绘制程序中计算草叶与碾压物 OBB 包围盒的求交。相交检测完成后,通过读取深度缓存来获得相交检测的结果。然后将发生相交的草叶数据重新组织成更小的纹理,分别与当前节点两个子节点的 OBB 包围盒进行相交检测,如此递归下去,直至达到需要的碰撞检测精度。读取深度缓存,得到被碰撞的草叶以及相应的碰撞位置。

由上述碰撞方法可见,当碰撞检测进行到包围盒树的叶子节点时,将叶子节点上的所有三角形与草叶进行相交检测,此时碰撞检测精确到碾压物体的每个面,粒度最高;当碰撞检测只进行到包围盒树的根节点时,演变成最基本的碰撞检测方法,此时粒度最低。根据实际应用的需要,我们可控制碰撞检测的精度,以达到所需的效果。

3.3 碾压物作用下草体形变计算

3.3.1 多边形草叶受力模拟

对于基于多边形表达的草叶,其在碾压物作用下发生形变的草体可分为两类:一类是直接受碾压物碰撞而变形的直接形变草叶;另一类是不与碾压物直接接触,但由于其它形变草叶的碰撞而产生变形的连带形变草叶。我们采用不同的策略得到上述两类草叶的受力情况,并以此为基础计算变形。

(1) 直接形变草叶受力

直接形变草叶的受力主要受到两个因素的影响:一是与草叶发生碰撞的节点的等效法线 EqN ,它从方向的角度反映了草叶受力情况;二是当前碾压物体的运动速度 v , v 越大草叶受力越大。因此,直接形变草叶的受力向量可表示为

$$EqF = |\vec{v}| \cdot EqN \quad (7)$$

等效法线与碰撞检测精度有着密切的关系。为了表征 OBB 层次包围盒中任一层次 l 中每个节点的包围盒对受碰草叶的影响,我们将第 l 层每个节点所包含面片的法线进行加权平均,求出该节点的等效法线,以面片的面积作为加权系数。设一个节点中的面片数为 $t(t=1,2,\dots)$,面片对应的面积是 $S_1, S_2, \dots, S_t$,对应的法线是 $N_1, N_2, \dots, N_t$,则该节点的等效法线为

$$EqN = \frac{S_1 \times \vec{N}_1 + S_2 \times \vec{N}_2 + \dots + S_t \times \vec{N}_t}{S_1 + S_2 + \dots + S_t} \quad (8)$$

通过求等效法线,将一个节点中的多个面片等效成一个面片。

要精确得到连带形变草叶的变形情况,需要在草叶间进行碰撞检测。为了减少计算开销,本文采用了一种类似于种系衍生的方法,将直接形变草叶作为祖先,将其受力向量向周围草叶传播,模拟出周围草叶的受力并将新求得的受力继续传播,直至所有草叶的受力都小于传播阈值。通过上述方法,确定所有连带形变草叶的受力情况。

定义 2(存活状态) 当草叶生命值大于给定传播阈值时,草叶处于存活状态。

定义 4(祖先) 碾压物发生碰撞的直接形变草叶。

定义 6(相邻关系) 对于任意草叶 G , 位于其上下左右方向的四片草叶与其存在相邻关系。

定义 7(平行关系) 对于第 I 代孩子 G , 如果与其相邻的草叶不是祖先, 也不是第 K 代孩子 ($K \leq I$), 则称它们之间存在平行关系。

假设第 M 代孩子 J 的受力向量为 EF_J , 生命值为 $|EF_J|$, 传播阈值为 L , 如图 5 所示, 则其繁衍过程如下:

Step1 判断 J 的生存状态。若为死亡状态,则停止繁衍,否则继续。

Step2 将 J 的受力向量 EF_J 正交分解为两个向量 EF_1' 和 EF_2' 。

Step3 取由向量 EF_1' 和 EF_2' 指向的且与 J 存在相邻关系的草叶, 判断它们是否与 J 存在平行关系。如果是, 则返回, 算法结束。否则, 两个草叶的受力向量分别为

式中, $att(at \in t[0, 1])$ 为衰减系数, 用于防止繁衍过程一直进行下去。

Step4 将两个草叶标记为第 $M+1$ 代孩子,繁衍结束。

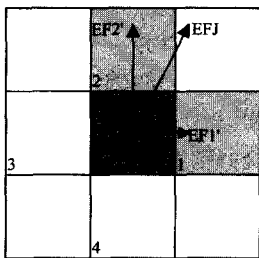


图 5 受力向量传递

3.3.2 草叶形变计算

(1) 基于多边形的草叶形变计算

通过碰撞检测计算,我们可以得到单根被碰草叶的碰撞节数以及草叶受力情况。根据以上信息,使用如下过程计算草叶形变。

Step1 计算偏移基向量

为保持变形过程中的连贯性及草叶整体稳定性,我们需要首先得到一个反映单根草叶整体形变趋势的基准偏移向量,控制草叶上各个顶点变形的偏移向量都以此向量为基准产生。基向量计算方法如图 6 所示。

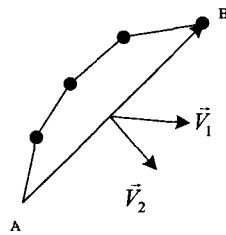


图 6 基向量计算

取草叶骨架线根部与末梢两点 A, B 连接形成向量 $\vec{AB}$, 该向量反映了草叶弯曲趋势方向, 草叶偏移基向量 $\vec{base}$ (反映草叶形变趋势) 方向应与 $\vec{AB}$ 垂直, 则偏移基向量可表示为

$$base = \frac{(\vec{V}_2 \cdot \vec{V}_1) \vec{V}_2}{num} \quad (10)$$

式中, $\vec{V}_2$ 为垂直于 $\vec{AB}$ 的单位向量, $\vec{V}_1$ 表示碰撞检测后得到的变形向量, num 为发生碰撞的节数。由式(10)可以看出, $base$ 向量方向与 $\vec{V}_2$ 相同, 大小与 $\vec{V}_1$ 在 $\vec{V}_2$ 方向上的投影大小成正比, 与碰撞节数成反比。

Step2 计算顶点偏移向量

为使草叶在变形过程中保持整体稳定性,避免出现非常规的扭曲现象,我们规定同节草叶顶点使用相同的偏移向量;整根草叶顶点使用方向相同的偏移向量,与偏移基向量 $\vec{base}$ 相同。此外,根据现实草叶弯曲情况,变形期间越靠近末梢的顶点偏移量越大,则各节草叶顶点对应偏移向量 $\vec{off}_i$ 为

$$\vec{off_i} = P_i \cdot \vec{base} \quad (11)$$

P_i 为各节的权值, 由权值函数 $P(x)$ 确定:

$$P(x) = \frac{1}{4}mx + ax^2 + bx + c \quad (12)$$

m 是第 1 节的初始权值, 变量 x 是对应的节数, 可以取 $x=1, 2, 3, \dots$ 得到对应的节数的权值, a, b, c 是常量。

Step3 草叶保长处理

直接利用上一步得到的各点偏移向量更新各节点位置可能会导致草叶拉伸的现象,因此,在将偏移向量作用于草叶节点时加入保长计算,如图7所示。

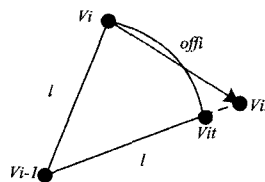


图 7 草叶保长处理

从根节点开始,逐步更新草叶各节顶点在偏移向量作用下的位置。第 i 节草叶顶点 V_i 在偏移向量 $\vec{off}_i$ 作用下得到新位置 V'_i 。显然, $\overrightarrow{V_{i-1}V'_i}$ 与 $\overrightarrow{V_{i-1}V_i}$ 长度不等,即单节草叶长度发生了变化。故我们以 $\overrightarrow{V_{i-1}V'_i}$ 为方向、草叶单节长度 l 为长度计算 V'_i ,即

$$V_{it} = V_{i-1} + L \cdot \overrightarrow{V_{i-1} V'_{it, normal}} \quad (13)$$

(2) 基于 billboard 的草丛形变计算

由于草丛 billboard 离视点较远,因此没必要对其进行精确的碰撞检测。我们用 billboard 的骨架线 m 表示 billboard,

计算 m 与碾压物体的 OBB 包围盒是否相交。若相交,则断定该 billboard 被碾压物体碾压,需要进行变形;否则不进行任何操作。billboard 形变计算如图 8 所示。

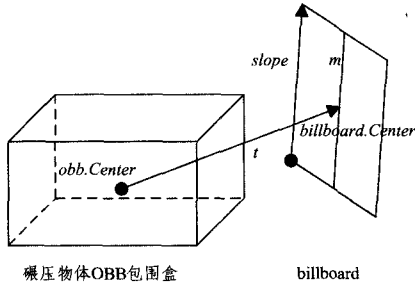


图 8 billboard 形变计算

在计算具体形变时,通过控制 billboard 的倾斜向量 $\vec{slope}^b$ 来展现碾压物体对草叶的影响,billboard 下部两个顶点固定,倾斜向量 $\vec{slope}^b$ 的计算方法如下

$$\vec{slope}^b = \vec{slope}_{origin}^b + \vec{t} \cdot R / |\vec{t}|^2 \quad (14)$$

式中, $\vec{t}$ 是从碾压物 OBB 包围盒中心到布告板中心的方向向量, R 表示 OBB 包围盒的对角线长度的一半。由式(14)可见,billboard 距离碾压物体越近,则倒伏程度越大,这与现实世界中的情况相符合。

3.3.3 层间动态过渡

针对不同草叶表达方法我们采用了截然不同的变形策略。为防止动画的不连续性,需实现层间的动态过渡。动态过渡存在两种情况:多边形草叶向 billboard 的过渡、billboard 向多边形草叶的过渡。

(1) 多边形草叶向 billboard 的动态过渡

我们根据多边形草叶数据来计算对应的 billboard 的 $\vec{slope}$ 值,从而实现多边形草叶向 billboard 的动态过渡。

如图 9 所示,某一时刻 patch 中第 i 根草叶的倾斜向量 $\vec{slope}_i^g$ 为

$$\vec{slope}_i^g = \vec{V}_n - \vec{V}_0 \quad (15)$$

草叶实时倾斜向量 $\vec{slope}_i^g$ 与其初始倾斜向量 $\vec{slope}_{origin}^g$ 之间的偏移向量为

$$\vec{off}_i^g = \vec{slope}_i^g - \vec{slope}_{origin}^g \quad (16)$$

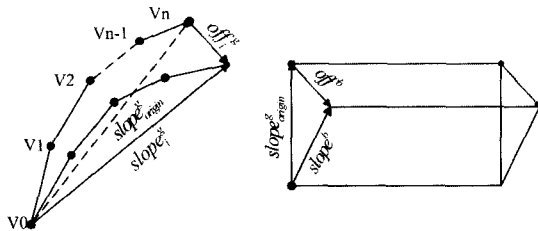


图 9 多边形草叶向 billboard 的动态过渡

计算 patch 中所有草叶的偏移向量并求取平均值,将其作为该 patch 所对应的 billboard 倾斜向量的偏移量。假设该 patch 上有 K 片草叶,则

$$\vec{off}^b = (\sum_{i=1}^N \vec{off}_i^g) / K \quad (17)$$

使用 $\vec{off}^b$ 更新 billboard 的倾斜向量

$$\vec{slope}^b = (\vec{slope}_{origin}^b + \vec{off}^b) \cdot \text{height}_g \quad (18)$$

式中, $\vec{slope}_{origin}^b$ 为初始布告板的倾斜向量,是竖直向上的向

量; height_g 为 billboard 的高度。根据计算得到的 $\vec{slope}^b$,改变 billboard 的上部两个顶点数据,可得到多边形草叶对应的 billboard 草丛变形。

(2) billboard 向多边形草叶的动态过渡

假设某一时刻 billboard 的倾斜向量为 $\vec{slope}^b$,相对于 billboard 的初始位置,其斜率向量偏移量 $\vec{off}^b$ 为

$$\vec{off}^b = \vec{slope}^b - \vec{slope}_{origin}^b \quad (19)$$

我们将 $\vec{off}^b$ 作为多边形草叶受力向量 $E_q F$,并规定其作用于草叶最末梢一节上,以此为基础计算多边形草叶形变,从而得到 billboard 所对应的多边形草叶姿态。

4 实验结果与分析

我们利用 OpenGL,C++ 实现了本文提出的草体碾压方法,实验的测试平台为一台配置为 Intel Pentium 2.6GHz CPU、2.0Gb 内存、NVIDIA 9800GT 显卡的微机。

我们建立了能展现 260,000 草叶数量的场景,单个 cell 包含 1024 根草叶,在场景中加入一个坦克模型,使其按固定的路径移动,与草叶发生碰撞,碰撞检测精度为 3,即所进行到的 OBB 层次包围盒树的层次为 3,视点位置固定,整个场景的绘制速度在 20fps 到 47fps 之间。

为了进一步验证碰撞检测精度对绘制效率的影响,在碰撞检测精度分别为 0,1,2,3,4,5,6 的情况下进行实验。各场景分别采样 5000 帧,获取最低帧速、最高帧速,并计算平均帧速。实验结果如表 1 所列。

表 1 不同碰撞精度下的绘制速度

精度	0	1	2	3	4	5	6
最低帧速(fps)	28	27	23	20	16	10	6
最高帧速(fps)	48	47	47	47	46	46	43
平均帧速(fps)	36	36	35	33	30	26	24

由表 1 可以看出,对于同一个碾压物,随着碰撞精度的提高,绘制速度会随之下降。但由于建模时采用了混合式草地场景表达方法,计算量得到控制,从而使平均帧速维持在较高水平。此外,实验发现当精度达到一定程度时,进一步提高精度对碾压效果真实感的提升并不明显,需根据实际情况(如碾压物模型复杂度、绘制速度要求等)选择合适的碰撞精度。与现有方法相比,本文方法可以提供更高精度的碰撞检测,真实感得到有效提升,同时场景整体绘制速度达到实时。在碰撞精度为 4 的情况下,最低帧速也有 16fps。

图 10 显示了不同碾压物作用下草体动态效果。

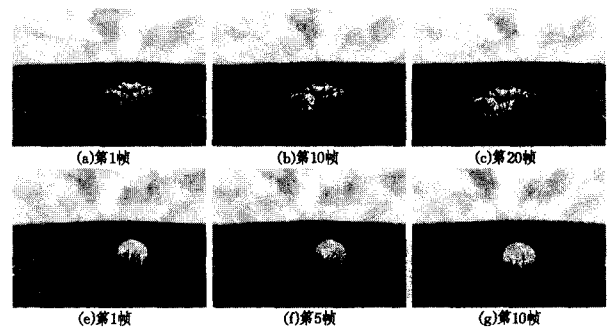


图 10 不同碾压物作用下草体动态效果

良好的改进潜力,利用其改进算法——SURF^[8]和PCA-SIFT^[9],可以进一步提高运行效率,对于摄像机视频场景的实时三维重建具有非常重要的意义。

参考文献

[1] Saxena A. Make3D: Learning 3D Scene Structure from a Single Still Image[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(5): 824-840

[2] Abdel-Aziz Y I, Karara H. MDirect linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry[C]// Proceedings Symposium on Close-Range Photogrammetry. 1971: 1-18

[3] Tsai R. An efficient and accurate camera calibration technique for 3D machine vision[C]// Proceedings of Computer Vision and

Pattern Recognition. 1986: 364-374

[4] 杨长江, 汪威, 胡占义. 一种基于主动视觉的摄像机内参数自标定方法[J]. 计算机学报, 1998, 21(5): 428-435

[5] Zhang Z. A flexible new technique for camera calibration[J]. IEEE-PAMI, 2000, 22(11): 1330-1334

[6] David G L. Distinctive Image Features from Scale-Invariant Key-points[J]. International Journal of Computer Vision, 2004, 60(2): 91-110

[7] 吴朝福. 计算机视觉中的数学方法[M]. 北京: 科学出版社, 2003

[8] Bay H. SURF: Speeded up robust features[C]// Computer Vision-ECCV 2006, PT 1, Proceedings. 2006, 3951: 404-417

[9] Ke Y, Sukthankar R. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors[C]// Proc. Conf, Computer Vision and Pattern Recognition. 2004: 511-517

(上接第 272 页)

此外,将本文方法绘制效果与文献[6,8]方法进行了对比,结果如图 11 所示。



图 11 不同方法绘制效果比较

由于文献[6]方法是根据一个受影响区域来判断草叶是否与碾压物发生碰撞,没有实际的碰撞检测计算,其结果过于粗糙,因此难以反映真实的交互效果。但该方法绘制速度较快,平均帧速达到了 35fps。文献[8]方法采用基于包围盒的碰撞检测,碰撞精度不高,平均帧速为 45fps。本文提出的可控精度草体碾压算法,在真实感和实时性间获得了较好的平衡,在碰撞精度为 4 的情况下,平均帧速达到了 30fps。

从草体表达方式、碰撞检测、碰撞物外形要求、GPU 加速、草叶间相互作用以及层间动态过渡等几个方面,将本文方法与已有方法进行了详细比较,结果如表 2 所列。

表 2 本文方法与其他方法比较

	草体 表达	碰撞 检测	碾压物 外形要求	GPU 加速	草叶间 相互作用	动态 过渡
Guerraz ^[6]	混合式	×	无限制	×	×	×
牛立新 ^[7]	粒子系统	低精度	球体	×	√	×
Zhao ^[8]	混合式	低精度	无限制	√	×	×
本文方法	混合式	可控精度	无限制	√	√	√

由表 2 看到,与文献[6-8]相比,本文方法提供了更高精度的碰撞检测,使模拟效果更加逼真,同时碰撞检测的精度可根据实际需要进行调控,具有较大的灵活性。其次,与文献[7]类似,本文方法也考虑了草叶间的相互作用,但我们并不进行叶间碰撞检测,而是采用受力向量传递的策略,提高了计算速度。此外,与现有方法不同,本文方法考虑了不同草叶表达方法间的动态过渡,有效消除了层间动画不连续的问题。

结束语 草地场景的真实感动态模拟一直是计算机图形学领域的研究热点和难点。本文提出了一种草体碾压效果模

拟方法,该方法采用混合式草地场景表达方法构造出不同细节的局部草地场景,利用基于 GPU 的碰撞检测算法、受力向量传递策略、billboard 斜率控制等手段,对不同类型的草体实施变形计算,有效克服了传统方法处理速度慢、模拟精度差以及对碾压物外形存在限制等缺陷,在视景仿真、娱乐游戏、军事仿真等领域具有较好的应用价值。

参考文献

[1] Boulanger K, Pattanaik S N, Bouatouch K. Rendering grass in real time with dynamic lighting[J]. IEEE Computer Graphics and Applications, 2009, 29(1): 32-41

[2] Wang Y M, Jiang X F, Ge L. Modeling and rendering of dynamic grassland scene in the wind[C]// Proceedings of the International Conference on Information Engineering. Washington DC: IEEE Computer Society, 2009: 69-74

[3] Habel R, Wimmer M, Jeschke S. Instant animated grass [J]. Journal of WSCG, 2007, 15(3): 123-128

[4] Bakay B. Animating and lighting grass in real-time[D]. Vancouver: University of British Columbia, 2003

[5] 王佳声, 唐好选, 孙振义. 自然风力作用下草体运动的模拟方法[J]. 哈尔滨工业大学学报, 2008, 40(3): 466-470

[6] Guerraz S, Perbet F, Raulo D, et al. A procedural approach to animate interactive natural sceneries[C]// Proceedings of the 16th International Conference on Computer Animation and Social Agents. Washington DC: IEEE Computer Society, 2003: 73-78

[7] 牛立新. 真实感图形算法与实时绘制技术研究[D]. 北京: 首都师范大学硕士学位论文, 2007

[8] Zhao X K, Li F X, Zhan S Y. Real-time animating and rendering of large scale grass scenery on GPU[C]// Proceedings of the 2009 International Conference on Information Technology and Computer Science. Washington DC: IEEE Computer Society, 2009: 601-604

[9] 范昭炜, 万华根, 高曙明. 基于流的实时碰撞检测算法[J]. 软件学报, 2004, 15(10): 1505-1514

[10] Philip J S, David H E. 计算机图形学几何工具算法详解[M]. 周长发, 译. 北京: 电子工业出版社, 2005