

采用虚拟粒子解决织物仿真中的碰撞问题

张 明,耿兆丰

(东华大学 信息科学与技术学院,上海 200051)

(mzhang_76@126.com)

摘 要:提出一种粒子系统,能在任何需要正确模拟碰撞的地方自适应的生成新粒子。当粗糙网格与物体发生碰撞时使网格自动细化。因此这种模型与一般的粒子系统相比,只需要少量的粒子就能获得精确的物理仿真效果。同时,使用粗糙网格进行快速仿真成为可能。

关键词:虚拟粒子;粒子系统;自适应碰撞;织物仿真

中图分类号: TP391.41 **文献标识码:** A

Using visual particles to solve the collision in textile simulation

ZHANG Ming, GENG Zhao-feng

(College of Information Sciences and Technology, Donghua University, Shanghai 200051, China)

Abstract: Particle systems that adaptively generate new particles was introduced, wherever this was required to model collisions correctly. When coarse meshes collide with an object they were automatically refined. Therefore this model allows physically accurate simulations that require a much smaller number of particles than regular particle systems. It is also possible for fast simulations with coarse meshes.

Key words: virtual particle; particle systems; collision adaptive; textile simulation

0 引言

弹簧质点粒子系统^[1,2]是模拟织物的一种比较恰当模型。粒子系统将可变形物体作为二维网格模拟,网格的顶点代表粒子。该模型默认所有质量平均分布在粒子上,粒子受到内、外力作用。其中,内力包括张力、弯曲力、剪切力,粒子之间通过虚拟弹簧连接并相互施加力的作用。

当可变形物体与环境另一个物体发生碰撞时,会发生两类碰撞:1)粒子与面碰撞;2)边与边碰撞。粒子系统中,力作用于粒子上。因此,对于可变形物体的粒子穿透环境时发生的碰撞,可以通过对穿透粒子施加力或更改其位置和速度来处理。然而,如何处理边、边碰撞或环境粒子穿透物体表面的碰撞却不那么明显。

由于发生这些碰撞,需要非常大的粒子数目(系统离散程度高),才能确保可变形物体精确的贴合所碰撞的环境表面。自适应是数值分析和几何建模中为避免均一、细致离散的一种常用概念。在此本文的目标同样是采用自适应方法降低物理建模的计算消耗。

实际仿真中发现在粒子系统的某些区域,粗糙的离散也能够实现精确仿真。然而,无论碰撞发生在何处,都要求可变形物体贴合其所碰撞的表面。正是这些发现促成了对在碰撞处生成新粒子的、碰撞自适应粒子系统的研究。

本文所提出的方法可以正确处理各种类型的碰撞,并且只需要较少数目的粒子就能获得良好的仿真效果。在此给出了织物与其环境中刚体的碰撞情况。自适应插入的粒子是仅受碰撞控制而不直接受微分方程控制的新粒子。这些新粒子称之为虚拟粒子。利用这种自适应性可以用非常粗糙的离散网格仿真织物,并向基于物理的实时模拟前进了一步。该方

法不依赖粒子系统的实际执行情况。

1 现有的研究工作

早期对可变形物体的模拟研究是基于使用连续模式的变形能量函数^[1]。文献[2,3]把粒子系统作为一种织物性能仿真的离散模型。文献[4]对计算机图形学所采用的织物建模技术进行了调查。文献[5]利用自适应性概念开发了适合高曲率的多解粒子系统,即当系统中的角度远远大于松弛角时采用高阶解。

近期研究的焦点是获得更好的仿真效果并保证实现动画所需的合理计算时间。文献[6]改善了数值性能,获得了相当大的时间步。他们提出的框架以约束而不是力的形式表述。然而,力可以从约束中求得。其主要贡献是运用隐式数值常微分方程求解器,单个求解步骤执行较快并能很有效的处理碰撞引起的约束。

碰撞检测和反应是物理建模的主要问题,文献[7]给出了一种碰撞检测的快速算法,但却没有针对不同类型的碰撞给出专门的解决方法。文献[8]描述了一些类型的碰撞,但也没有涉及边与边碰撞的处理问题。

通常是靠限制织物和环境粒子之间的最小偏差距离,来阻止那些未被系统正确处理的碰撞。但如果使用的粒子数目较少时,最小距离将增大,并且由粗糙织物网格所覆盖的环境物体形状会被错误描绘。甚至会出现这样的情况,当织物下落到物体上时,由于没有任何阻力,将会逐渐下落,最终穿透物体。

2 虚拟粒子的添加和计算

通过对粒子施加力或改变粒子位置和速度^[3,8,6],确保织

收稿日期:2005-09-01;修订日期:2005-11-10

作者简介:张明(1976-),男,山东青岛人,博士研究生,主要研究方向:计算机图形学、纺织CAD;耿兆丰(1938-),男,江苏张家港人,教授,博士生导师,主要研究方向:智能控制、控制工程、计算机辅助设计。

物粒子在环境之外。然而,边的碰撞会引起穿透。这种错误是由于对可变形模型求解不充分造成的。可以通过向系统添加新的粒子来修正。向描述可变形物体的固定网格中添加粒子,会导致每个时间步粒子数量的增加,而且 t_n 时刻的粒子在 t_{n+1} 时刻可能不再需要,或者根据该时刻的碰撞情况,这些粒子应当位于其他位置。

因此不需要从微分方程中计算出这些粒子的轨迹,但要从每个时间步的碰撞中计算出这些虚拟粒子的位置。它们仅对当前时间步有效。所以这些虚拟粒子的位置由当前织物和环境的位置控制。在某一时间插入虚拟粒子后,这些粒子被用于显示当前帧,从而用户不会觉察到模拟物体的任何穿透。

由于插入了虚拟粒子,模型不匹配最初的物理描述,边的长度增加了,即织物在张力能没有增加的情况下伸长了。虚拟粒子对其邻近粒子施加力的作用。每个时间步生成新的网格拓扑并计算该拓扑的受力或者能量。注意不必计算作用在虚拟粒子上的力,因为这些粒子不在微分方程的轨迹上运动。但虚拟粒子施加力将邻近粒子拉回到一起,抵消由于碰撞引起的伸长。因此,把虚拟粒子集成到模型中提高了物理精确度。

该方法假定所有基础粒子都在环境网格之外,即该方法与传统的碰撞检测和反应协同阻止基础粒子穿透环境。仅在传统的碰撞反应之后,才计算每个面的虚拟粒子。环境中的物体也需要以网格形式给出。这并非是一种限制,因为实际上所有几何物体都可以网格形式给出,或可转换为网格。在此使用三角形织物网格,计算所有虚拟粒子。如果织物以矩形网格形式给出也可很容易的转换为三角形网格。

目标是确保所有自适应改进的织物网格的边在环境网格之外。这可通过一种投影方法获得,该法中所有环境的边或者边的片断都被投影到织物上。那些穿透织物的环境粒子以及边的片断被插入最初的织物网格拓扑中。

把环境网格的边映射到织物网格上的投影 P ,是通过定义投影 P 到一个织物网格面 f 的约束 P_f 给出的。假定所有面的法向量 \mathbf{n}_f 为标准的。首先,为每个面 f 定义一个投影中心:

通过三个平面 S_i 为每个面 f 定义边界体积,在此 S_i 是把 f 面与邻近的织物面 f_i 分开的平面。 S_i 由法向量 $\mathbf{n}_{av_i} = \frac{\mathbf{n}_f \times \mathbf{n}_{f_i}}{\|\mathbf{n}_f \times \mathbf{n}_{f_i}\|}$ 确定,该法向量定义为与两个面的法线和公共顶点 P_i 垂直:

$$S_i: \langle \mathbf{n}_{av_i}, (\mathbf{X} - P_i) \rangle \geq 0$$

令 C_f 为这三个分割平面的交点。 P 到面 f 的约束是中心投影,由其投影中心 C_f 给出。如此定义投影则面上方的空间被边界体积分割。

如果法向量 \mathbf{n}_{av_i} 线性相关,那么 C_f 是在无穷远处的一点而且中心投影变为平行投影。在任一情况下投影 P_f 由齐次坐标表示的投影空间中的一点定义。

定义一个投影,把边 e 映射到 f 张成的平面上。因为要限制中心投影到单个网格面 f ,并且只考虑实际穿透环境的部分织物网格,所以必须采取一些裁剪步骤。

环境边 $e = \overline{XY}$ 投影到顶点为 P_1, P_2, P_3 的面 f 上步骤如下:

1) 计算 X 和 Y 到 f 的距离:

$h_X = \langle \mathbf{n}_f, \mathbf{X} - P_1 \rangle, h_Y = \langle \mathbf{n}_f, \mathbf{Y} - P_1 \rangle$ 。只考虑那些高度为正值的边上的点。高度为负值的点不会穿透。如果一条边的一个片段高度为负值则裁剪掉,并通过线性插值计算新的端点。如果 $h_X < 0$ 和 $h_Y < 0$ 边根本不会穿透则跳过。

2) 给定 C_f 将 e 投影到 f 张成的面上。

3) 投影边与面 f 的边求交。

裁剪掉面 f 外的片段。包含在面 f 内的片段的端点记为 \bar{V} , \bar{W} (所有投影点都加横线表示)。可能发生如图1所示的四种情况。如果 f 内不包含边的任何片段则忽略该边,如图1(a)。图1(b)中边被裁剪了两次。在图1(c)中只计算一个新端点,而在图1(d)中整条边都包含在 f 中。

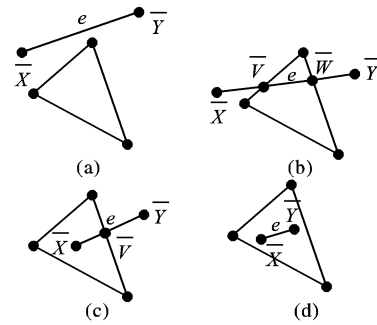


图1 边 e 相对网格面 f 的四种可能情况

4) 如果 \bar{V} 不是投影边 e 的一个端点,则要计算 $V = P^{-1}(\bar{V})$ 。

令 α, β 为 V 相对于 X, Y 的重心坐标。 $(X, Y, V) = \frac{\alpha}{\beta}$ 在仿射变换下是不变的, V 可以通过线性插值计算。然而,在这样一种中心投影的情况下, V 不能通过线性插值计算,因为比率在投影变换下是变化的,而交叉比是不变的:

$$\frac{\text{ratio}(X, Y, V)}{\text{ratio}(X, Y, A)} = \frac{\text{ratio}(\bar{X}, \bar{Y}, \bar{V})}{\text{ratio}(\bar{X}, \bar{Y}, \bar{A})} \quad (1)$$

如果有辅助点 A 和它的像 \bar{A} 则可以用来计算 V 。因此将辅助点 $A = \frac{1}{2}(X + Y)$ 投影到面 f 上得到 \bar{A} ,并求解(1)式得到 V 。

5) 投影 P 下,从线段端点到其像的射线不能与其他物体相交。例如,如果织物没穿透,而是位于环境物体的反面就会发生这种情况。从环境物体对面发出的射线将穿透环境对着可变形物体的那个面。

织物的每个面 f 变成包含最初三个顶点(基础粒子)以及所有新计算出的虚拟粒子和连接这些粒子的边的子网格。因为 P_1, P_2, P_3 和所有虚拟粒子 V_i 的位置均为已知,所以在这个阶段该子网格的几何属性为已知。通过带约束的德洛奈三角剖分计算拓扑,为此要用到 \bar{V}_i 。通过以正确的顺序插入位于面边界中 f 上的新顶点获得新的边界拓扑。其重心的一个分量相应于 P_1, P_2, P_3 为零,因此重心可以用来给位于边界上的顶点分类。

面内的线段和面边界被限制到子网格拓扑。在此采用了 P_1, P_2, P_3 和所有位于平面 f 内的 \bar{V}_i 的带约束的德洛奈三角剖分(运行时采用了文献[11]提出的一种非常稳定的三角剖分)。然后将二维计算得到的拓扑与三维几何相结合,给出子网格,即用 V_i 代替 \bar{V}_i 。最后所有子网格(面)合并为包含所有基础粒子和虚拟粒子以及连接它们的边的单一网格。

一个简单的碰撞例子如图2所示。四个粒子的系统与盒子的右上边发生边/边碰撞。为解决这些边/边碰撞,在织物网格中插入三个虚拟粒子。

因为当所有的边投影到所有面上时,投影方法具有 O (环境边数与织物面数之积)的复杂度,所以使用加速结构是必要的。为了减少必须进行投影的边/面对,采用与可变形物体碰撞检测相同的算法^[7]。

在实现上采用基于网格的加速方法。可变形物体的面合并为一个网格。然后为每条边选择包含这些边的网格单元。因此仅需要对那些在这些网格单元里的织物面进行投影。由于三角剖分是局部的,且其计算消耗受到插入一个面的最大粒子数目的限制,所以三角剖分的复杂度为 $O(\text{织物面数})$ 。

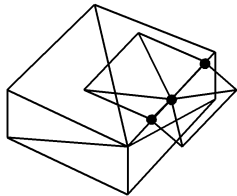


图2 四个粒子的网格落到盒子上

3 物理交互的虚拟粒子

前面已经提到虚拟粒子不仅用于视觉效果,还对邻近粒子施加力的作用。

如果上一节计算的三角形网格用于物理仿真,一个粒子的邻近粒子数目每个时间步都会发生变化。由于拉伸、剪切和弯曲产生的内力分别依赖于粒子和粒子间的距离、平面内的剪切角和平面外的弯曲角。系统处于平衡状态时的松弛距离和松弛角也必须符合新的拓扑和几何结构。

研究时发现,进行物理仿真时只在表面边界处添加虚拟粒子就足够了,可以忽略那些位于网格内的虚拟粒子。因此不必处理变化了的拓扑结构,只需局部上改变粒子的位置和速度即可。

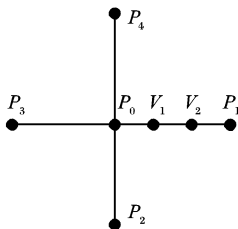


图3 粒子 P_0 的局部拓扑结构

下面用例子解释一下:

如图3所示,假定粒子 P_0 有四个近邻 P_1, P_2, P_3, P_4 和两个虚拟粒子 V_1, V_2 在边 P_0, P_1 上计算。作用于 P_0 上的力由 V_1 代替 P_1 作为 P_0 的直接近邻计算。

因此在实现上只需用 V_1 的位置和速度替换 P_1 的即可。因为假定织物中该点将附着在环境中的那一点上,所以虚拟粒子的速度需要计算阻尼力并设置为对应环境点的速度。静态环境中,所有虚拟粒子的速度设置为零。

同理,虚拟粒子也可以替代另外三个基础粒子 P_2, P_3, P_4 。

为控制张力必须计算新的弹簧松弛长度。当一条边被分成几段时,这些片断的松弛长度之和,须等于最初边的松弛长度以确保面积不变。因此第 j 个片段的松弛长度 r_j 由

$$\frac{r_j}{r} = \frac{\Delta l_j}{\sum \Delta l_i}$$

计算,在此 r 是这条边的松弛长度, l_i 是第 i 个片断的当前长度。

这样每次调用评价函数检查每个粒子的虚拟邻居。如果找到基础粒子的虚拟邻居代替原始邻居并由力引导粒子至正确方向然后恢复正确的边长。因此虚拟粒子的插入只是简单的几何坐标替换。由于基础粒子中间存在虚拟粒子,弯曲力

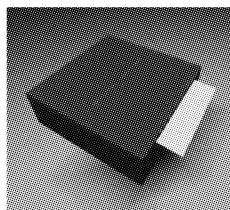
和剪切力不能从一个基础粒子传递到另一个上,所以不能用虚拟粒子计算这些力。这样做会令系统解耦。

虚拟粒子的插入并未改变材料性能。假定将粒子间的弹簧一分为二。使用上述新的松弛长度,并未改变材料性能。然后移动虚拟粒子解决碰撞,这样在仿真效果更逼真的同时,保持了正确的材料参数。

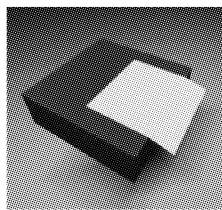
4 实验结果

在此给出了一种处理织物仿真碰撞问题的方法,并且使用少量粒子就能确保快速仿真计算。

图4给出了仅由两个三角形片模拟的织物落在盒子上的情形。图4(a)所示穿透是由于三个边/边碰撞引起的。如图2所示的三个虚拟粒子插入到边界上,因此新的网格没有嵌入盒子里。



(a)未使用虚拟粒子



(b)使用虚拟粒子

图4 四个粒子的网格落在盒子上

如图5给出了在粒子系统基础上,采用本文所述技术进行仿真的结果。在此采用了隐式积分方法求解常微分方程。尽管只采用了100个粒子,但也能准确的表现出桌布的形状。

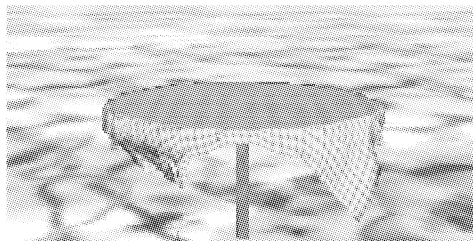


图5 圆桌上桌布的悬垂仿真

另外一个例子如图6所示,一块布落在一个球上。由于球的网格相当复杂,包含1000多条边。如果不采用虚拟粒子,那么当粗糙网格的织物落到上面时,球的顶部就会显露出来。图6给出的结果通过插入虚拟粒子,使粗糙的织物网格完全盖住了球表面。

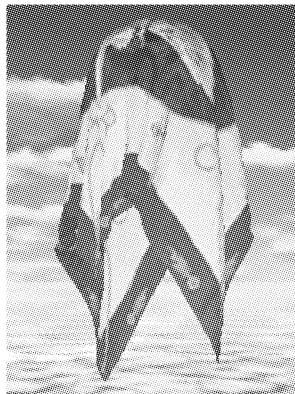


图6 织物落在球上的效果仿真

自适应的额外计算消耗是由于虚拟粒子的计算造成的。

(下转第493页)

ACU 注册监听器,在有特殊情况发生时,可以指挥注册的监听器更改流程,设定审批流转具有很好的适应性;

4) 基于委托代理模型的审批流转方案对于临时要求,例如:重复审批、同时进行多个 ACU 等具有良好的支持。

3 委托代理模型的实现

从上述委托代理模型的介绍可以看出,它的实现主要由五个部分构成:

1) 节点定制:所谓审批就是某一单据或文档在人与人之间传递,给出最终的决定。因此很容易想到节点就是某个具体的人。而在实际工作中,人员的变动频度是比较高的,如果在定制委托代理模型时将节点定制成人员,在以后系统运作过程中,一旦人员变动,模型就要调整,这是不合理的。所以通常将节点定制成人员组、职务、角色等相对稳定的信息,而在指定流转的具体单据或文档时再选定某一组、职务或角色中的具体的人员。

2) 事件状态对象库:ACU 内部在没有发生特殊情况时是按照 ACU 定制时指定的流转流程进行处理,不涉及特殊处理。而系统中的每一特殊情况都对应一个自定义事件状态对象类。在 Java 这样的面向对象的程序设计语言中都提供了对自定义消息、事件的支持。在 Java 语言中可以通过继承 java.util.EventObject 生成自己的事件状态对象类。

3) 监听器接口库:与定义的事件状态对象相对应,创建相应的监听器接口。在 Java 中,事件操纵方法都被定义在继承了 java.util.EventListener 类的 EventListener 接口中,根据事件状态对象库中定义的各种事件信息定义生成监听器接口库。

4) 监听者库:监听者是相应监听器接口的具体实现,是一个个的 Java 类,每个监听者都实现了具体的某个监听器接口,也就是给出了事件的响应。

5) ACU 定制:AC 是最简单的 ACU,ACU 实质上是由节点集、控制方式(线性、并行、协商)构成。而 ACU 定制除生成 ACU 单元外,还要建立 ACU 与事件监听者间的事件流,也就是在 ACU 单元中必须为事件监听者提供注册和注销的方法。在 Java 中,指定节点、控制方式,形成 ACU 单元,然后在 ACU 单元的类中定义一个储存事件监听者的 Vector 数组,接着在 ACU 单元类中添加 add $\times \times$ Listener 和 remove $\times \times$ Listener 方法即可。

要建立基于委托代理模型的审批流转系统,除上述提到的主要构成部分外,还涉及具体系统中要进行审批流转的对象,此处不再进一步展开。

4 结语

本文在分析目前一般审批流转方案不足的基础上提出了一个委托代理模型,该模型具有一定的通用性,可用于电子单据、文档等对象的审批流转,对审批流转类系统的设计与实现有一定的参考价值。

参考文献:

- [1] 陶晓东,潘浙丹. 基于多代理技术的申请审批系统[J]. 计算机工程, 2003(20): 191 - 193.
- [2] 高志伟,牛江川,张利,等. Internet 环境下政府并联审批系统研究[J]. 计算机系统应用, 2004, (3): 12 - 15.
- [3] HORSTMANN CS, CORNELL G. Java 2 核心技术(卷 I: 基础知识)[M]. 程峰,黄若波,章恒翀,译. 北京: 机械工业出版社, 2003.
- [4] <http://info.shangdu.com/new/2003-5-8/200358110522.htm>, 2005.
- [5] <http://www.matrix.org.cn/resource/article/0/264.html>, 2005.

(上接第 490 页)

在仿真过程中,只需给出新的坐标和松弛长度。表 1 中第三列给出了执行投影步的时间,第四列是计算拓扑结构的时间,最后一列是总的时间。这些值都是在一秒的仿真时间内在 PIV/2.0GHz 处理器上的平均时间。

表 1 每秒仿真时间内计算虚拟粒子所用的平均时间

实例	粒子数目	投影时间(ms)	拓扑计算时间(ms)	共计(ms)
圆桌	100	38.5	38	76.5
球	400	150.5	83.5	234

因为只使用了位于边界上的虚拟粒子,所以要为物理仿真计算新的拓扑结构。整个执行情况依赖于基本的物理模型和使用的数值求解器。进一步的工作主要集中在投影算法的改进上。更加优化的加速方法和算法改进将进一步减少投影的时间。

参考文献:

- [1] TERZOPOULOS D, FLEISCHER K. Deformable models[J]. The Visual Computer, 1988, 4: 306 - 331.
- [2] BREEN DE, HOUSE DH, WOZNY MJ. Predicting the drape of woven cloth using interacting particles. Proceedings of SIGGRAPH'94 (Orlando, Florida, July 24 - 29, 1994)[A]. Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH[C]. ACM Press. ISBN 0-89791-667-0, 1994. 365 - 372.
- [3] EBERHARDT B, WEBER A, STRASSER W. A fast, flexible,

- particle-system model for cloth draping[J]. IEEE Computer Graphics and Applications, 1996, 16(5): 52 - 60.
- [4] NG HN, GRIMSDALE RL. Computer graphics techniques for modelling cloth[A]. IEEE Computer Graphics and Applications[J]. 1996, 16(5): 52 - 60.
- [5] HUTCHINSON THD, PRESTON M. Adaptive refinement for mass/spring simulations[A]. In 7th Eurographics Workshop on Animation and Simulation[C]. 1996. 31 - 45.
- [6] BARAFF D, WITKIN A. Large steps in cloth simulation[A]. SIGGRAPH 98 Conference Proceedings, Annual Conference Series[C]. ACM SIGGRAPH, Addison Wesley, ISBN 0 - 89791 - 999 - 8, July 1998. 43 - 54.
- [7] VOLINO P, MAGNENAT - THALMANN N. Efficient selfcollision detection on smoothly discretized surface animations using geometrical shape regularity[J]. In EuroGraphics, 1994, 13: 155 - 166.
- [8] PROVOT X. Collision and self-collision handling in cloth model dedicated to design garments. In Graphics Interface'97[A]. Canadian Information Processing Society, Canadian Human-Computer Communications Society[C]. 1997. 177 - 189.
- [9] SHEWCHUK JR. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. Applied Computational Geometry: Towards Geometric Engineering, volume 1148 of Lecture Notes in Computer Science[A]. From the First ACM Workshop on Applied Computational Geometry[C]. 1996. 203 - 222.