第 34 卷　第 2 期　　计 算 机 工 程　　2008 年 1 月
Vol.34　No.2　　Computer Engineering　　January 2008
·多媒体技术及应用·　文章编号：1000—3428(2008)02—0215—02　文献标识码：A　中图分类号：TP18

带阴影布料的实时模拟

张新峰，沈一帆
(复旦大学计算机科学与技术系，上海 200433)

摘　要：介绍了如何使用弹簧-质点系统对布料进行模拟，虽然该方法速度较快，但是会出现数值计算不稳定的现象。该文提出了隐欧拉方法，结合 shadow volume 算法，解决了柔软物体的实时阴影生成问题。该方法运行速度较快，生成的阴影边缘齐整，视觉效果较好。
关键词：物理模拟；弹簧-质点系统；实时阴影；阴影体方法

Real-time Cloth Simulation with Shadow

ZHANG Xin-feng, SHEN Yi-fan
(Department of Computer Science and Technology, Fudan University, Shanghai 200433)

【Abstract】This paper introduces how to simulate the behavior of cloth using a spring-mass system. The method runs efficiently, but has mathematical instability. Combined with shadow volume algorithm, this paper proposes an implicit Euler method, overcomes the problem of real-time shadow for soft object. The method runs very fast, the edge of the generated shadow is very smooth, and visual effect is perfect.
【Key words】physical-based simulation; spring-mass system; real-time shadow; shadow volume method

随着计算机硬件运算速度的提高，人们想要获得更为真实的实时场景，对于像布料这样的柔软物体的模拟是必不可少的。一般来说，织物覆盖了人体的 90%，高真实感的织物模拟能够创造出真实的场景。因此，对于织物甚至为一般柔软物体模拟的研究，一直是图形学界研究的热点。要做到照片级的真实渲染，复杂的光学渲染引擎是必不可少的，但是现有的家用电脑很难达到实时全局光照的要求，因此，在局部光照下采用“迂回”方法，可以生成逼真的全局光照的效果。实时阴影生成，尤其是对于柔软物体的阴影，算法的响应速度必须要快，本文利用了阴影体(shadow volume)方法来生成比较真实的实时阴影。该方法运行速度较快，而且生成的阴影边缘齐整,但是难以直接扩展到非多边形的物体上去。

1　织物的建模

Hing[1]指出，早期对于织物建模主要有：

(1)几何方法

用几何方法(geometry techniques)对织物建模时，并没有真正地考虑织物本身的物体特性，而仅仅是用一些几何方程去模拟织物的皱褶和各种折痕。所用的几何方程是悬链线方程为

$$y = a\cosh(x/a)$$

其中，cosh 是双曲余弦函数；a 是缩放因子。

根据数学物理方程的结论，悬链线是物体势能最小的状态，因此，这种方法本质上是能量最小原理的一种应用。

(2)物理方法

该方法主要是基于材料力学的研究成果，将要模拟的织物表示为三角片或长方形网格，然后根据微分方程的数值解法求解。

该方法根据能量最小化原理提出了一种模拟悬挂着的布料的方法，将能量表示为

$$E(P_{i,j}) = k_s E_{elasti,j} + k_b E_{bendi,j} + k_g E_{gravi,j}$$

其中，k_s, k_b, k_g分别是弹力、弯曲、密度常数；$E_{elasti,j}$, $E_{bendi,j}$, $E_{gravi,j}$分别是弹性势能、弯曲势能、重力势能。

(3)混合方法

该方法混合了物理方法和几何方法。该混合方法(hybrid techniques)主要利用几何方法作为物理方法中需要求解的微分方程的一个初始条件，然后用数值方法不断地求精。

上面所介绍的方法各有优缺点，几何方法直观、速度较快，但没有考虑物体本身的性质，模拟结果不是很自然。

早期的物理方法虽然模拟结果较好，速度适中，但是会出现数值不稳定的现象，Baraff[2]采用隐式欧拉方法求解微分方程，得到了良好的效果，但运算速度会大幅度地下降，在织物稍大的情况下，基本上不能达到实时要求。

2　实时阴影

要想达到使场景真实目的，实时阴影是必不可少的，实时阴影的算法主要有 2 种：

(1)阴影映射(shadow mapping)[3]。从光源往物体看过去，如果当前要渲染的点能被光源看到，那么该点就不在阴影里；否则，如果该点不能被光源看到，也就是说该点和光源之间有遮挡物，那么该点就在阴影里。这种方法需要先将光源作为观察点，将从光源出发能够达到的最近点的深度值记录在一张深度纹理里(depth texture)，因此，其占用显存比较大。

(2)阴影体(shadow volume)[4]。指的是一块多边形区域，该区域是从光源看被物体遮挡的区域。通过计算任意渲染点的阴影体顺时针和逆时针的面数就能够知道该点是否处在阴影里，从而可以将真正的阴影渲染出来。

基金项目：国家自然科学基金资助项目(60473104)
作者简介：张新峰(1982－)，男，硕士研究生，主研方向：计算机图形学；沈一帆，教授、博士
收稿日期：2007-03-02　**E-mail：**042021174@fudan.edu.cn

3 织物的物理模型

本文主要基于物理模型对布料进行建模，建模的方法主要是弹簧质点系统(spring-mass system)。该系统运算比较简单，更新速度比较快，在质点足够多的情况下面效果也比较真实。图 1 为用这种方法生成的布料，可以看到该方法生成的布料能够逼真地模拟出真实布料的褶皱(buckling)。

图 1　布料物理模拟效果

3.1　弹簧质点系统

根据广义虎克定理为 $\delta = E\varepsilon$。其中，δ 为应力；E 为弹性系数；ε 为应变。

在发生挤压的情况下，物体内部某一微小质元对周围质元的弹性内力与该质元所发生的形变成正比。

为了简化，可以将内力施力质元和受力质元分别考虑，施力质元为弹簧，受力质元为质点，并作如下假设：(1)弹簧是没有质量的。(2)每个弹簧两端都连接一个质点。(3)质点是没有体积的。(4)每个弹簧都有一定的原始长度。(5)每个质点都有一定的质量。

3.2　系统结构

模拟的布料是长方形的，弹簧和质点的连接关系见图 2。

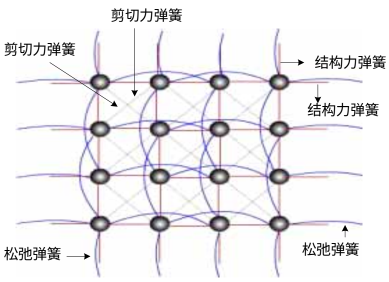

图 2　弹簧质点系统结构

3 种弹簧如下：

(1)结构力弹簧(structural spring)，连接横向和纵向的质点，将整个布料的结构支撑起来。

(2)剪切力弹簧(sheer spring)，连接对角线和反对角线方向的质点，提供布料的剪切内力，生成更真实的褶皱。

(3)松弛弹簧(flexion spring)，连接横向和纵向隔一个质点的质点，避免布料的自相交。

3.3　数值计算问题及其求解

如果将第i行、第j列的质点的质量记为$m_{i,j}$，则可以记质量矩阵$\boldsymbol{M}$=[$m_{i,j}$]，假设其在t时刻的位置为 $x_{i,j}^{t}$，记其位置矩阵为$\boldsymbol{X}$=[$x_{i,j}^{t}$]，同样记其受力矩阵为$\boldsymbol{F}$，其中，每个元素 $f_{i,j}^{t}$ 为t时刻，第i行、第j列质点所受的力，则根据牛顿第二定理有

$$\ddot{x} = \boldsymbol{M}^{-1}\boldsymbol{F} \tag{1}$$

将这个微分方程用显示欧拉方法展开可以得到($\boldsymbol{M}$ 与时间无关，为常数矩阵)：

$$F_k^t = k(l_k^{t-1} - l_k) \tag{2}$$

$$a^t = \boldsymbol{M}^{-1}\boldsymbol{F}^t \tag{3}$$

$$v^t = v^{t-1} + a^t\Delta t \tag{4}$$

$$x^t = x^{t-1} + v^t\Delta t \tag{5}$$

其中，F_k^t 为第 k 个弹簧在 t 时刻对周围 2 个质点的弹力大小；l_k^{t-1} 为第 t-1 时刻第 k 根弹簧的长度；l_k 为第 k 根弹簧的原长，Δt 为时间步长，为一固定值。

这样给定初始条件$x0$, $\boldsymbol{M}$, L就能求出各个时间点的x_t了。

3.4　数值稳定性问题

上面所述的方法是针对求解微分方程的，当弹簧的弹性系数比较大时，容易出现数值不稳定，甚至出现发散情况。在这种情况下，由于始终不能达到里平衡，因此模拟出来的布料会莫名其妙的“抖动”。

为了解决这种问题，可以利用了隐式欧拉方法。用显示欧拉方法来求解微分方程，主要的假设是，t~t+1 内，质点的受力情况始终不变，但这个假设明显与实际情况不符，这是造成显示欧拉方法数值计算不稳定的主要原因。也是显示欧拉方法跟隐式欧拉方法的主要区别。

隐式欧拉方法的主要计算方法是：假设物体t~t+1 内所收的加速度是变化的，设t时刻质点的加速度为a_t，t+1 时刻的加速度为a_{t+1},则上可以用$(a_t+a_{t+1})/2$ 来代替这段时间的加速度。

但是，a_{t+1}是跟t+1 时刻的质点的位置相关的，而这是未知的，隐式欧拉方法用预测-迭代的方法来确定这个t+1 时刻的状态。该问题可以描述如下：

假设x_t, v_t, a_t为t时刻物体的位置、速度、加速度，求t+1时刻物体的加速度为a_{t+1}。

用隐式欧拉方法的求解如下：

(1)用显示欧拉方法，求解出 t+1 时刻质点位置的猜测值 x_{t+1}^0 和加速度的猜测值 a_{t+1}^0。

(2)猜测质点在 t 时刻到 t+1 时刻的加速度为

$$a_{t+1}^1 = \frac{a_t + a_{t+1}^0}{2}$$

根据这个值计算 x_{t+1}^1。

重复步骤(1)、步骤(2)，直到计算出来的 x_{t+1} 趋于稳定。

4　阴影与碰撞

4.1　阴影体方法

本文所用的实时阴影算法主要是阴影体方法。该方法的主要思路是阴影里的点必定是处在阴影体正面和反面之间的，如果一个像素点落在阴影体的正面和反面之外，则该点必定不是阴影点。

该方法的伪代码如下：(1)从光源出发，找出物体的轮廓(silhouette)。(2)将光源与 silhouette 连接并无限延长形成阴影体。(3)绘制阴影体的正面，将所有画到的像素的 Stencil Buffer 值加 1。(4)绘制其背面，将所有画到的像素的 Stencil Buffer 值减 1。(5)根据 Stencil Buffer 的值，画一个大的矩形匡，达到足够将整个屏幕覆盖。

这个方法最重要的地方在于如何找出从光源看出去物体的轮廓(silhouette)。

定义从光源出发，能够看到的面为可见面，不能看到的面为不可见面。那么对于任一个多变形来说，轮廓上的边必定有这样的性质：该边所相邻的两个面，必定一个为可见面，一个为不可见面。因此，找可见面的伪代码如下：

对于每一条边，有：　(下转第 219 页)