浙江大學

硕 士 研 究 生 读 书 报 告

题 目 《Efficient and Accurate Collision Response for Elastically Deformable Models》读书报告

作者姓名 张慧兰

作者学号 21951164

指导教师 李启雷

学科专业 软件工程

所在学院 软件学院

提交日期 二〇 一九 年 十二 月

The Effect Of The Requirements Analysis On The System Design

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Li Qilei

By

Zhang Huilan

Zhejiang University, P.R. China

2019

1 问题描述

基于物理的模型现在广泛应用于许多计算机图形应用程序，如动画电影和视频游戏。典型的动画由许多(复杂的)对象组成，这些对象可以彼此交互，也可以与环境交互。这些物体的行为基于它们的内部动力学，通过一些材料模拟(如刚体、布或流体模拟)来建模。然而，物体的每一个动作也会影响场景中其他物体的运动。因此，对碰撞事件的精确处理极大地提高了仿真的真实感。除了平移和旋转运动外，可变形物体的运动也受到形状变化(变形)的影响。由于变形一般是未知的，所以无法直接计算出碰撞的准确时间和位置。此外，碰撞反应会影响物体的动力学，从而影响它们的运动，理想情况下，它还应该考虑到精确的摩擦。

处理接触问题的典型方法是将其重新表述为速度约束，并使用拉格朗日乘数来表示未知的接触力大小。然后求解混合线性互补问题(MLCP)，得到既符合碰撞响应问题又符合模型动力学的解。约束问题通常通过将其重新表述为线性互补问题(LCP)，使用(预计)高斯-赛德尔方法来解决。LCP 矩阵的建立需要广义质量矩阵的逆。广义质量矩阵可以直接用于刚体仿真计算。然而，对于可变形的物体，这种逆过程不是直接可用的，必须近似。因此，人们常常寻求一个近似值或解决一个耦合问题系统。另外，直接求解原 MLCP 是一种可行的可变形模型求解方法。Ramage 和 Wathen 在 1994 年的论文中比较了求解由 Stokes 方程的有限单元离散化引起的耦合不定问题的共轭剩余法(CR)和使用两个嵌套共轭梯度(CG)求解器的两级方法的性能。结果表明，在这类约束条件下，CR 方法优于 CG 方法。受他们的结果启发，作者研究如何扩展 CR 方法来有效地解决包含不等式约束的接触问题。对于耦合刚体和可变形体的稳定模拟，重要的是在每一步结束时都要完全解决所有的碰撞。然而，当碰撞被解决时，产生的变形可能导致新的碰撞或其他地方接触力的变化。这经常导致一种接触与实际几何形状不一致的状态。如果不纠正这些错误，这些触点最终将把能量引入系统，导致振荡和不稳定。此外，变形越大，这种不匹配出现的可能性越大。即使对于刚体，这种不匹配也会发生。为了最小化这些误差，需要解决非线性接触问题。

2 问题解决

算法由三部分组成，以下给出算法伪代码：

（1） 碰撞处理系统

ALGORITHM 1: Pseudo-code of our collision handling system.

1 Discretize computational domain;
2 Initialize BVH and additional data structures;
3 **while** *Simulating* **do**
4 Update matrix $\mathbf{A}$ using FEM (see Equation (4));
5 Find all potential collision candidates (Section 5.1);
6 Linearize and check all active constraints;
7 $\mathbf{r}_0 = \mathbf{d} - \mathbf{B}\mathbf{y}_0$; $\mathbf{p}_0 = \mathbf{C}^{-1}\mathbf{r}_0$, $j = 0$;
8 **while** *Not converged* **do**
9 $\alpha = \frac{\mathbf{r}_j^T \mathbf{C}^{-1}\mathbf{B}\mathbf{C}^{-1}\mathbf{r}_j}{\mathbf{p}_j^T \mathbf{B}\mathbf{C}^{-1}\mathbf{B}\mathbf{p}_j}$;
10 $\mathbf{y}_{j+1} = \mathbf{y}_j + \alpha\mathbf{p}_j$;
11 $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha\mathbf{B}\mathbf{p}_j$;
12 **if** $\alpha > 0$ **then**
13 EvaluateConstraints($\|\mathbf{r}_{j+1}\|$, j, ϵ_1, ϵ_2);
14 **if** *No constraints changed* **then**
15 $\beta = -\frac{\mathbf{r}_{j+1}^T \mathbf{C}^{-1}\mathbf{B}\mathbf{C}^{-1}\mathbf{r}_{j+1}}{\mathbf{r}_j^T \mathbf{C}^{-1}\mathbf{B}\mathbf{C}^{-1}\mathbf{r}_j}$;
16 $\mathbf{p}_{j+1} = \mathbf{C}^{-1}\mathbf{r}_{j+1} - \beta\mathbf{p}_j$;
17 $\mathbf{B}\mathbf{p}_{j+1} = \mathbf{B}\mathbf{C}^{-1}\mathbf{r}_{j+1} - \beta\mathbf{B}\mathbf{p}_j$;
18 **else**
19 $\mathbf{r}_{j+1} = \mathbf{d} - \mathbf{B}\mathbf{y}_{j+1}$;
20 $\mathbf{p}_{j+1} = \mathbf{C}^{-1}\mathbf{r}_{j+1}$;
21 $\mathbf{B}\mathbf{p}_{j+1} = \mathbf{B}\mathbf{C}^{-1}\mathbf{r}_{j+1}$;
22 **if** $\left(\|\mathbf{r}_{j+1}\| < \epsilon_1 \wedge \|\mathbf{U}\mathbf{r}_{j+1}\|_\infty < \epsilon_2\right) \vee \|\mathbf{B}\mathbf{C}^{-1}\mathbf{r}_{j+1}\| < \epsilon_1$ **then**
23 EvaluateConstraints($\|\mathbf{r}_{j+1}\|$, 0, ϵ_1, ϵ_2);
24 **if** *No constraint states have changed* **then**
25 Check all candidates for *new* collisions (Section 5.2);
26 **if** *No new constraints are added* **then**
27 Re-linearize constraints (Section 3.3 and Equation (21));
28 **if** *No constraints are updated* **then**
29 Advance simulation;
30 **break**;
31 Recompute residual (see lines 19–21);
32 $j = j + 1$;

（2） 约束分析

ALGORITHM 2: Pseudo-code for constraint evaluations.

```
Function EvaluateConstraints(r, i, ε1, ε2):
    interval = max(1, ⌊(log²(r/ε1))/3⌋);
    if i mod interval == 0 ∨ i == 0 then
        foreach contact k do
            if Non-penetration constraint active then
                if j_k v − c_{k,n} ≥ 0 and λ_k ≤ 0 then
                    Deactivate constraint, λ_k = 0;
            else if j_k v − c_{k,n} ≤ 0 then
                Activate constraint, set friction to kinetic friction;
            if Non-penetration constraint active then
                if Friction constraint in static friction mode then
                    if ‖γ_k‖ > μλ_k and (j_{k,t} v − c_{k,t})^T γ_k > 0
                    then
                        Switch to kinetic friction;
                        δ̂_k = γ̂_k, λ'_k = ‖γ_k‖/μ, γ_k = 0;
                    else if (j_{k,t} v − c_{k,t})^T δ̂_k ≤ 0 then
                        Switch to static friction;
                        γ_k = γ'_k, γ'_k = 0;
                    UpdateKineticFriction();
                if Z^{-1} S_d [Δλ_k, Δγ_k^T + Δγ'^T_k]^T > ε2 then
                    Use updated constraint k and report change to
                    solver;
                else
                    Discard changes for constraint k;
        d = [(b − J_t γ')^T, c_n^T, c_t^T]^T /*Update RHS*/;
```

（3） 动摩擦更新

ALGORITHM 3: Pseudo-code for kinetic friction update.

```
Function UpdateKineticFriction():
    if Constraint k in kinetic friction mode then
        if ‖γ'_k‖ ≠ μλ_k then
            Compute Simple Moving Average λ_{k,a};
            λ'_k = λ_{k,a};
        if 0 < δ̄_k^T (j̄_{k,t} v − c̄_{k,t}) < cos(ε_θ) /*If angle θ > ε_θ*/ then
            Δδ_k = (j̄_{k,t} v − c̄_{k,t}) − δ̄_k /*Compute difference*/;
            δ̄_k = δ̄_k + α_θ Δδ̄_k /*Decreases θ*/;
        γ'_k = μλ'_k δ̄_k /*Update friction force*/;
```

3 创新与短板

创新在于以下四点：

（1） 不需要计算 Delassus 算子

（2） 基于 CR 方法，提供变形和接触力的估计

（3） 提出了一个简单而有效的预调节器，可以显著减少迭代次数和计算时间。

（4） 该方法考虑了较大的变形和接触力，速度更快，仿真复杂性增强。

短板在于必须使用定制的冲突检测系统，而 ICA 和 SP 可以使用任何高度优化的可用系统。由于原文的方法是非常准确的，它对由于错误的检测碰撞发生而产生的冲突约束包容性较小。这对碰撞检测方法提出了额外的要求，这样就不会产生不正确或冲突的约束。此外，原文的方法中的摩擦处理可能不允许简单地替换本文提议的基于 CR 的求解器，例如，用路径求解器。

4 研究应用

可以应用于许多计算机图形应用程序，如动画电影和视频游戏。有许多可以彼此交互或与环境交互的对象，行为基于内部的动力学，并且在碰撞时会引发变形。对碰撞事件的精确处理可以极大地提高仿真的真实感。