快速布料的方向约束执行模拟

Oktar Ozgen 和 Marcelo Kallmann

摘要：我们引入了一种新的方法，大大改善了实时布料模拟系统中经常使用的防止过度拉伸的迭代边缘长度约束执行。 我们的方法是基于约束的方向性强制和从固定的顶点开始，在重力方向上传播同时逐步扫描布边。 所提出的方法成功地检测到要被校正的有意义的弹簧，忽略了对整体视觉效果没有任何意义的弹簧。 所提出的方法简单，有较好鲁棒性，能够实现逼真的布料模拟而不会出现过度拉伸，不引起任何视觉伪影，并且显着降低约束执行过程的计算成本。

关键词：物理仿真，布料仿真，约束执行

1 介绍

布料模拟被广泛用于计算机图形学，并且许多系统能够实时模拟中等复杂的布料模型。在实现可变形表面的几种可能方法中，基于粒子的系统仍然是实现交互式实时结果的最流行模型。基于粒子的布料模型中的一种常见不足之处是过度拉伸的影响，并且通常采用特定的几何边缘长度来执行程序。

与显式积分方法相结合的约束的迭代强制适用于具有适度颗粒数的布料模拟系统。这种组合成功地避免了求解方程组的需求，并因此能够进行在计算上快速和在视觉上令人满意的模拟，适合于计算机游戏。尽管研究已经发展到一般的约束执行问题，但是自从 Provot [15]提出以来，有用的迭代几何边长执行方法并没有得到改进。我们在本文中提出了一种新的方法，大大改善了这种方法。我们的方法实施简单，具有良好的鲁棒性，能够实现逼真的布匹行为，没有任何视觉伪影，提高了常规迭代约束执行的计算时间达 80％。

所提出的方法基于通过考虑拉伸发生最多的方向来计算边缘校正的有意义的单程序。首先，我们通过实验确定不同类型的考虑弹簧的修正优先级，以便最小化引力的拉伸效应。然后，我们从给定的模拟中的所有固定顶点开始进行修正，同时向布料的底部扩展。我们也限制每个顶点的校正数。通过优先考虑不同类型的弹簧的修正顺序，并将所有顶点的修正数量限制在一个很小的数量上，我们的方法成功地检测到了要修正的重要弹簧，忽略了对整体视觉效果没有任何意义的弹簧。我们的最终方法是能够显着降低实施过程约束的成本，同时成功消除了不必要的过度拉伸效应。

2 相关工作

布料模拟是计算机图形学中的一个中心话题，并且有丰富的文献资料

[5,4,15,8,21,3,10,6,19,18]可以引用。我们重点回顾以前的工作如何使用约束执行。

Provot [15]提出了迭代约束执行过程的常用方法。 在这种方法中，以反复的方式扫描布料，并且识别所有相对于其静止长度而言超过给定阈值的所有弹簧。 这些弹簧随后通过沿着弹簧轴线移动连接到弹簧的两个颗粒而恢复到它们的静止长度，如果弹簧被拉伸，则使它们更靠近; 或者如果弹簧过度压缩，则彼此远离（参见图 1）。

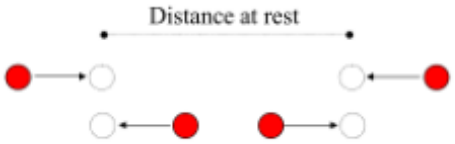

图1 两个由弹簧连接的粒子的校正方向

以上描述的过程的结果是产生了更硬的布料行为。但是请注意，并非所有的弹簧都可以被完美地修正，因为一个弹簧修正可能会改变先前已经修正的弹簧。这种蛮力纠正方法有一些固有的附加限制。对布料的一个纠正性迭代并不总是保证在视觉上令人满意的结果，此外，布料可能在几个区域呈现出振荡行为。为了克服这些限制，常见的方法是多次应用校正程序，直到获得视觉上满意的结果。显然，增加约束执行迭代的次数会降低仿真的性能，并且没有可靠的方法来确定实现期望的结果需要多少次迭代。在文献[15]中提到弹簧的修正顺序完全取决于它们所使用的数据结构，并且在全局范围内将约束条件扩展到整个布料对象的情况下，弹簧的顺序可能会更加重要，应该加以研究。这一观察结果是为我们提出的定向约束实施方法提供了主要动力。

除了迭代方法之外，还可以通过求解一个非线性约束方程组来解决这个问题。Terzopoulos 等[19]用 GaussSeidel 方法用线性方程组近似解[19]。另一种流行的方法叫做简化坐标表达式。在这个公式中，一个给定自由度数的无约束系统受到一系列约束条件的限制，从而消除了一些自由度，从而导致广义坐标[11]。拉格朗日乘子的公式也广泛使用，并且比定义的速度和其他不能使用简化坐标方法[1,2]制定的类型的约束更有优势

3 布料模型

我们的布料模型的网格结构是基于[6]提出的方案。 布表示为四边形粒子网格。粒子通过无质量弹簧相互连接。 有三种类型的弹簧元件，它们负责拉伸，剪切和弯曲力。弹簧的连通性如下描述：由 p(i，j)索引的粒子通过拉伸弹簧连接到由 p(i±1，j)，p(i，j±1)。 剪切弹簧将粒子连接到 p(i±1，j±1)，弯曲连接粒子到 p（i±2，j)，p(i，j±2)和 p(i±2， j±2)。

3.1 力

系统的动力学由牛顿的第二运动定律 F = ma 决定，其中 m 是质量，a 是粒子的加速度。基于施加到粒子上所有外部和内部的力量的合力 F，在每个时间步长计算加速度。

引力由 $F_g = mg$ 给出，它是我们系统中唯一的外力。 作用在粒子上的内力是由与每个粒子相连的弹簧，弯曲和剪切弹簧产生的力。 胡克定律给出了两个粒子之间的弹簧力 F_s：

$$F_s = -k_s(|l| - r)\frac{l}{|l|} \quad (1)$$

其中 $l = x_i - x_j$ 是两个粒子位置之间的差值，r 是弹簧的静止长度，k s 是线性弹簧常数。

3.2 Verlet 算法集成

虽然可以将约束执行过程与各种集成方法相结合，但是我们在这项工作中使用了 verlet 集成方法。Verlet 一体化是一种源自分子动力学领域的数值方法。由于其简单性，性能和稳定性，它已成为实时布料模拟的流行，特别是在电脑游戏中。

Verlet 方法将每个粒子的当前位置和前一个位置存储为系统的状态。因此，速度由位置隐式表示。每个粒子的位置更新步骤计算如下：

$$x_{t+1} = 2x_i - x_{t-1} + ah^2 \quad (2)$$

其中 x_{t+1}，x_t 和 x_{t-1} 是下一个当前和前一个时间步长的粒子位置，a 是当前影响粒子的加速度，h 是积分时间步长。 影响粒子的加速度是根据作用在其上的总力来计算的。系统的阻尼可以通过改变等式中的常数倍数（值 2）来微调。 减小到小于 2 会增加阻尼; 而增加到 2 以上会减小阻尼。

由于速度是隐式计算的，所以即使使用较大的时间步长，该方法也趋于保持稳定。但是，尽管模拟仍然是快速和稳定的，使用大的时间步长仍然会导致位置超调并导致超弹性的布料。 因此，依赖于 Verlet 积分的模拟经常与迭代约束执行过程耦合，该过程试图恢复布料模型中由弹簧连接的每对粒子之间的原始距离。

4 方向约束执行

我们的定向约束执行方法通过弹簧校正的有意义的遍历顺序实现了改进的结果。我们的方法通过防止冗余重复纠正来最小化所需的全部纠正的数量，并避免通常由简单的迭代约束执行遍历导致的振荡行为。

我们的方法是基于这样的观察结果：在给定的模拟场景中，弹性最大的弹簧倾向于是受接触力影响最大的区域。 因此，织物表面上最拉伸的区域被期望是靠近布的固定顶点的区域，这些顶点是不会相对于它们各自的碰撞物体移动的顶点。 通过定义首先校正布料固定顶点周围的弹簧的遍历顺序，并且扩大对重力方向的校正，可以确保校正是有效的而没有冗余。

我们的算法假定布料在重力下延伸并具有一组固定的顶点。 这些固定的顶点对应于将布贴在人物身上的顶点，对旗杆或桌布的旗子等等。由于可能有多于一个固定顶点来开始校正展开，所以重要的是 同步源自不同固定顶点的校正。 我们的实验表明，未

能同步扩展会导致严重的视觉伪影; 同时扩展仍然是该算法的一个重要方面。

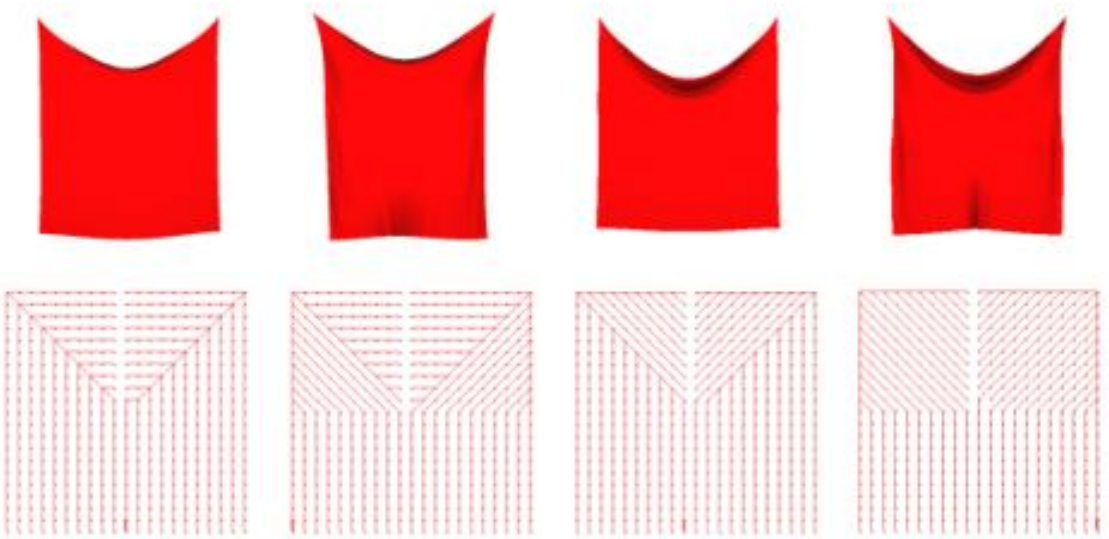

图2：在不同类型的弹簧中选择不同的修正顺序得到的修正模式及其各自的变形结果。

根据不同类型的弹簧所选择的顺序来分析布料的整体视觉外观也是至关重要的。如果我们将不同类型的弹簧及其伸长方向分类，我们可以提出三个主要类别：1）向左或向右伸展的拉伸弹簧（水平拉伸弹簧），2）朝布料的底部伸长的拉伸弹簧（垂直弹簧）和 3）伸向布料底部的剪切弹簧。请注意，我们不考虑向上延伸的弹簧，以确保修正将朝着重力方向扩展。

随着处理所确定的不同类型的弹簧的顺序，可以设计出与该顺序相关的边缘遍历过程。 算法 1 总结了我们的最终边缘校正排序生成过程。 该算法的细节在下面的段落中给出。

算法 1 接收固定顶点 V_f 的集合作为输入。 在算法的开始，我们使用一个空的队列 Q 来存储保持校正扩展的同步顺序。 当算法终止时，List L 为填充正确的边缘顺序，以便在运行期间进行校正。

整个算法的进行过程如下：首先将所有的固定顶点推入展开队列 Q 中，然后在队列中没有顶点的情况下逐个展开和展开。 过程 Get Horizontal Stretch Springs (v s)返回连接到顶点 v s 的水平拉伸弹簧。 类似地，过程 Get Vertical Stretch Springs (v s)返回连接到顶点 v s 的垂直拉伸弹簧，并且过程 Get Shear Springs (v s)返回连接到顶点 v s 的剪切弹簧。 最后，过程 Get Target Vertex（v s，s）返回由弹簧 s 连接到顶点 v 的顶点。

Algorithm 1 Ordered Edge Correction

Compute Correction List (V_f)

1. L ← null
2. for all v such that v ∈ V_f do
3. Q.push(v)
4. end for

5. while Q is not empty do
6. $v_S \leftarrow$ Q.pop()
7. $S_h \leftarrow$ Get Horizontal Stretch Springs (v_s)
8. for all s such that s $\in$ S_h do
9. $v_g \leftarrow$ Get Goal Vertex (v_s ,s)
10. Expand (L,v_s ,v_g)
11. end for
12. $S_v \leftarrow$ Get Vertical Stretch Springs (v**s**)
13. for all s such that s $\in$ V_s do
14. $v_g \leftarrow$ Get Goal Vertex (v_s ,s)
15. Expand (L,v_s ,v_g)
16. end for
17. $S_s \leftarrow$ Get Shear Springs (v_s)
18. for all s such that s $\in$ S_s do
19. $v_g \leftarrow$ Get Goal Vertex (v_s ,s)
20. Expand (L,v_s ,v_g)
21. end for
22. end while
23. return L

对于每个顶点展开，我们首先找到所有水平拉伸弹簧连接到它。然后，我们通过

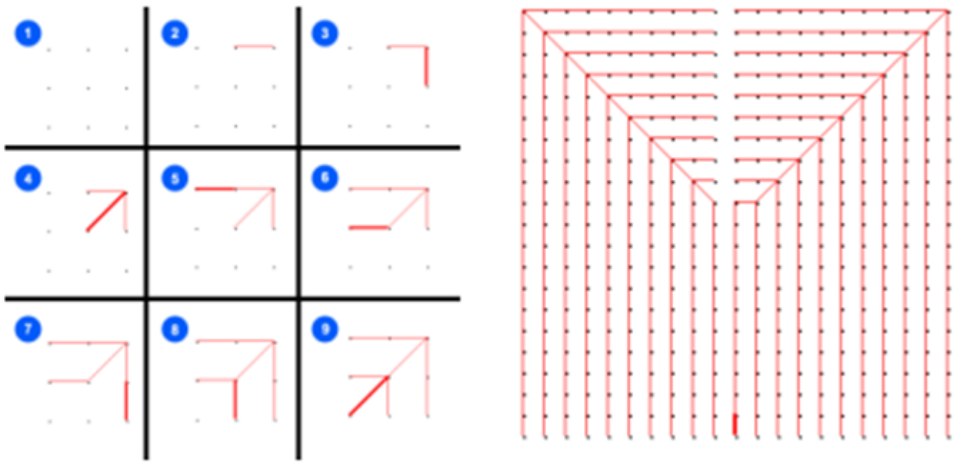

图3：左：从布料右上固定顶点开始的逐步校正顺序。右：从两个顶角固定的布料创建的校正图。

这些弹簧找到顶点所连接的目标顶点。最后，我们尝试向这些顶点展开修正图。当调用 Expand 程序时，不能保证目标顶点的扩展。该算法首先检查目标顶点已被访问了多少次。如果它已经达到访问限制数量，那么我们不扩大。如果访问限制还没有达到，则算法扩展到目标顶点，并将当前顶点到目标顶点的边添加到有序校正 List L.目标顶点的访

问次数将递增，目标顶点将会被添加到扩展队列 Q 中。对于垂直拉伸弹簧和剪切弹簧，这些过程也将被重复。当队列中没有顶点时，算法将停止。在算法结束时，我们将完成有序的校正 List L。

ListL 是预先为在实时模拟期间要考虑的每组固定顶点预先计算的。 然后，在仿真中的每个积分步骤之后，迭代约束执行步骤将遍历 List L 中的边缘，仅校正每个边缘中的第二顶点的位置，从而满足每个边缘的期望长度。在图 3 中展示使用该算法获得的示例结果。

4.1 使用多个修正地图(索引)

我们的方法依赖于预先计算的固定排序（List L 中编码），以便实时最大化计算性能。由于固定顶点的集合可能在模拟期间改变，所以预先计算的列表 L 的有效性可以根据变形事件的不规则性和不可预测性而有所降低。 我们在各种场景中测试了我们的方法，例如与实心球相互作用的平面布以及与执行“时装表演”的女性模型相互作用的复杂布模型。 在简单和复杂的两种情况下，所获得的各种布物碰撞和布料碰撞似乎都不会导致获得的结果质量的显着下降。

我们的方法还可以预先计算几个校正图，以进一步提高动画的性能和准确性。 在动画中重复出现的对应于不同碰撞状态的预先计算的地图可以根据模拟中的事件在它们之间切换来使用。 例如，在穿着长裙的行走角色的动画中，角色的膝盖与裙子发生的隐形事件交替地摆动，对边缘校正列表产生影响。

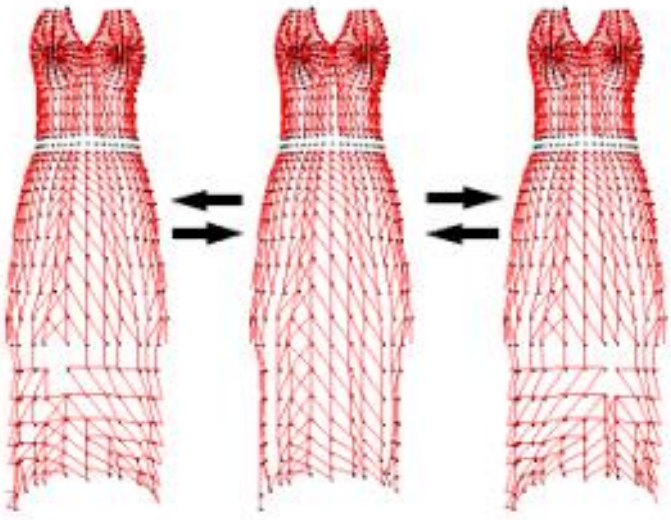

图4：行走时长裙模拟使用的三个交替校正图：左膝图（左图），无膝关节图（中图），左膝图图（右图）。

每当膝盖碰到裙子时，碰撞检测模块都会纠正碰撞的顶点， 我们也可以把它们当作固定顶点在额外的校正列表中。 对于 1）膝盖不接触裙子，2）只有右膝盖接触裙子，3）只有左膝盖接触裙子的情况，我们得到了改善的结果。 如图 4 所示，每次我们检测到膝盖碰撞（或碰撞），我们切换到相应的修正图。 如果变形是针对一个给定的步行动画特定的，则这些事件可以在动画的时间参数化中被编码，消除基于碰撞检测事件的决

定。 因此，多个校正图的使用适合于可靠的循环动画例如步行的模拟

5 结果与讨论

如前所述，我们的修正方法取决于固定顶点集合的可预测性。 在布点模拟场景中，一组固定点变化非常频繁，我们的方法将需要在每次显着变化之后更新校正图。 在这种情况下，一个用于检测校正 ListL 的有效性并在每次固定顶点集合从初始集合显着变化时触发排序更新的自动过程可以被集成到仿真系统中。但是，这仍然保持模拟整体性能而带来的的负面影响。 另一方面，如果固定顶点集上的变化不是很频繁，那么在整个计算时间内，这并不是一个减速的主要原因。

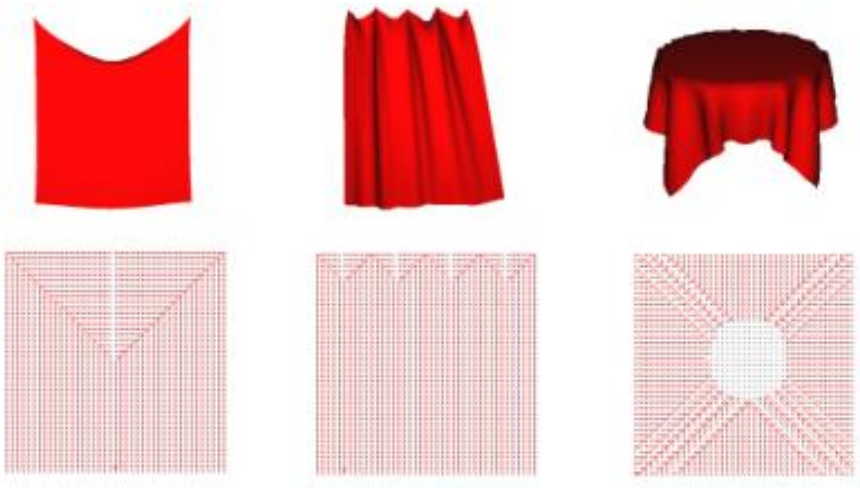

图5：该图显示了由我们的算法生成的校正图及其相关的布料模拟场景。

我们还假定布料在一个主要恒定的外力（重力）下拉伸。 虽然这个假设涉及绝大多数的布料模拟场景，但更复杂的情况下拉伸方向受到多种变化的外力的影响尚未经过测试。

我们的修正方法取决于布表面上的一组固定顶点。 在大多数布料模拟场景中，例如挂衣服，斗篷，旗帜，桌布或穿戴的角色，在整个动画期间固定顶点的集合保持不变。因此在这种情况下，校正图只能在非常少的时间内预先计算一次。 对于 2.4GHz 英特尔酷睿 2 双核计算机上的 1681 粒子组成的布料，修正图的重新计算时间约为 0.52 秒。

就视觉质量而言，我们的模型通过在边缘列表上进行单个校正通过而获得视觉上满意的结果。 这是对原来的（无序的）约束执行程序的一个重大改进，这个程序依靠多次通过才能取得好的结果。 我们的方法的另一个重要的优点是定向校正完全消除了在多次通过程序中经常看到的不稳定的布料行为。

5.1 不足之处

如前所述，我们的修正方法取决于固定顶点集合的可预测性。 在布点模拟场景中，

一组固定点变化非常频繁，我们的方法将需要在每次显着变化之后更新校正图。 在这样的情况下，可以在仿真系统中集成用于检测校正列表 L 的有效性并在每次固定顶点集合从原始集合显着变化时触发排序更新的自动过程。 但是，这仍然会保持惩罚模拟整体性能的负面影响。 另一方面，如果固定顶点集上的变化不是很频繁，那么在整个计算时间内，这并不是一个主要的减速。

我们还假定布料在一个主要恒定的外力（重力）下拉伸。 虽然这个假设涉及绝大多数的布料模拟场景，但更复杂的情况下拉伸方向会受到多种变化的外力的影响尚未经过测试。

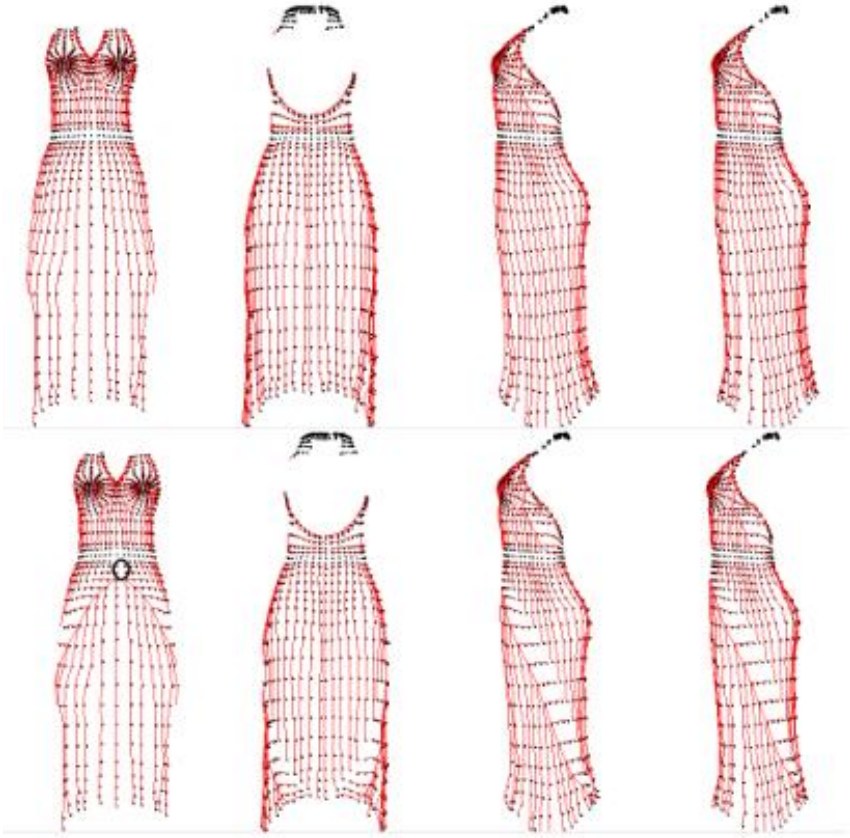

图6：最上一行：在肩部和腰带区域选择固定顶点的修正顺序。 底行：将少量顶点（由左下角图像上的圆圈表示）添加到固定顶点集合中。 注意获得的新的校正顺序是如何不同的，以及校正图如何在视觉上对应于布的预期屈曲。

6 结论

在布料模拟中，我们引入了一种方法来确定迭代边长执行的有效反向次序。 我们的方法是健壮的，实施简单的，并能够实现逼真的硬布行为，而不会造成过度伸缩或视觉伪影(如摇晃的效果)。此外，我们的方法提高了正则迭代约束执行程序的计算时间 80%。我们相信，我们的方向性强制执行方法将证明其本身对于使用基于粒子的可变形模型，

特别是在计算机游戏等实时应用中的许多场景有用。

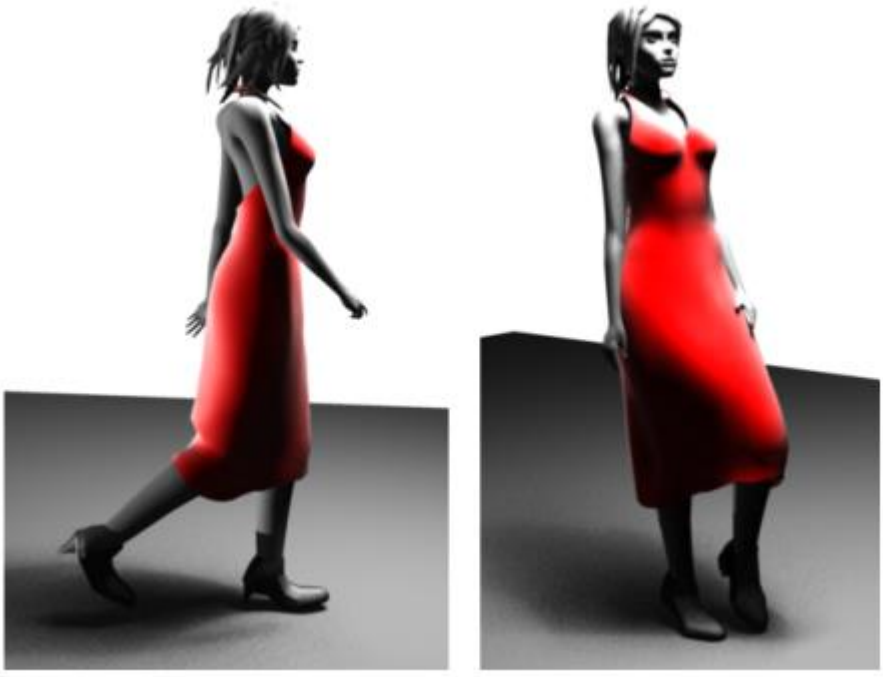

图7:一个长裙的例子，如果没有强制执行的话，这个长裙会明显地过度延伸。 我们的方法以有效的方式消除了过度修复。

致谢：我们要感谢 Robert Backman 在场景渲染和随附视频的准备方面提供的宝贵帮助。

参考文献

1. Baraff, D.: Linear-time dynamics using lagrange multipliers. In: Computer Graphcis Proceedings, Annual Conference Series. pp. 137–146. New York, NY, USA (1996)
2. Baraff, D., Witkin, A.: Physically based modeling: Principles and practice. In: ACM SIGGRAPH ’97 Course Notes (1997)
3. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proc. of SIGGRAPH’98. pp. 43–54. New York, NY, USA (1998)
4. Breen, D., House, D., Wozny, M.: Predicting the drape of woven cloth using interacting particles. In: Proc. of SIGGRAPH’94. pp. 365–372. ACM, New York, NY, USA (1992)
5. Carignan, M., Yang, Y., Magenenat-Thalmann, N., Thalmann, D.: Dressing animated synthetic actors with complex deformable clothes. In: Proc. of SIGGRAPH’92. pp. 99–104.ACM, New York, NY, USA (1992)
6. Choi, K., Ko, H.: Stable but responsive cloth. In: Proc. of SIGGRAPH’02. pp. 604–611.ACM, New York, NY, USA (2002)
7. Desbrun, M., Schröder, P., Barr, A.: Interactive animation of structured deformable objects.In: Proc. of Graphics interface. pp. 1–8. San Francisco, CA, USA (1999)
8. Eberhardt, B., Weber, A., Strasser, W.: A fast, flexible, particle-system model for cloth

draping. In: IEEE Computer Graphics and Applications'96. pp. 52–59. IEEE (1996)
9. Hauth, M., Etzmuß, O., Straßer, W.: Analysis of numerical methods for the simulation of deformable models. The Visual Computer 19(7-8), 581–600 (2003)Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: Exploration of bagging ensembles comprising genetic fuzzy models to assist with real estate appraisals. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 554–561. Springer, Heidelberg (2009)
10. L House, D.H., Breen, D.E. (eds.): Cloth modeling and animation. A. K. Peters, Ltd., Natick,MA, USA (2000)
11. Isaacs, P.M., Cohen, M.F.: Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In: Computer Graphcis Proceedings, Annual Conference Series. pp. 131–140. New York, NY, USA (1987)
12. Ling, L.: Aerodynamic effects. In: House, D.H., Breen, D.E. (eds.) Cloth modeling and animation, pp. 175,195. A. K. Peters, Ltd., Natick, MA, USA (2000)
13. Ling, L., Damodaran, M., Gay, R.K.L.: Aerodynamic force models for animating cloth motion in air flow. The Visual Computer 12(2), 84–104 (1996)
14. Müller, M., Schirm, S., Teschner, M., Heidelberger, B., Gross, M.: Interaction of fluids with deformable solids: Research articles. Computer Animation and Virtual Worlds 15(3-4), 159–171 (2004)
15. Provot, X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior.In: Proc. of Graphics Interface '95. pp. 147–155. New York, NY, USA (1995)
16. Robinson-Mosher, A., Shinar, T., Gretarsson, J., Su, J., Fedkiw, R.: Two-way coupling of fluids to rigid and deformable solids and shells. ACM Transactions on Graphics (Proc. of SIGGRAPH'08) 27(3), 1–9 (2008)
17. Selle, A., Su, J., Irving, G., Fedkiw, R.: Robust high-resolution cloth using parallelism,history-based collisions, and accurate friction. Transactions on Visualization and Computer Graphics 15(2) (2009)
18. Terzopoulos, D., Fleischer, K.: Deformable models. The Visual Computer 4(6), 306–331(1988)
19. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. In: Proc. of SIGGRAPH '87. pp. 205–214. ACM, New York, NY, USA (1987)
20. Tu, X., Terzopoulos, D.: Artificial fishes: physics, locomotion, perception, behavior. In: Proc.of SIGGRAPH '94. pp. 43–50. ACM Press, New York, NY, USA (1994)
21. Volino, P., Courchesne, M., Thalmann, N.M.: Versatile and efficient techniques for simulating cloth and other deformable objects. In: Proc. of SIGGRAPH'95. pp. 137–144. ACM,New York, NY, USA (1995)

本文译自： Oktar Ozgen ， Marcelo Kallmann，et al．Directional Constraint Enforcement for Fast Cloth Simulation：In Proceedings of The Fourth International Conference on Motion in Games (MIG), 2011，276 (12) :424-435．International Conference on Motion in Games

一种无布块快速布料的碰撞解决算法

Suejung B. Huh 和 Dimitris N. Metaxas

摘要：布料动画是计算机图形学中的一个重要应用领域，但由于布块问题，迄今为止，一种具有多条皱纹的快速移动布料一直难以生产。布块是布料的分块不自然聚集的冻结区域，是制作逼真的布料动画的障碍。因此，我们提出了一种新颖的布料碰撞分辨率算法，可以防止布料在快速运动中形成团块。我们的解析算法的目标是使布料快速移动，而不会有任何不自然的冻结布块，同时防止在模拟的任何时刻布料之间的穿透以及刚体与布料穿透。无论布料运动的速度如何，布料的非穿透状态都能保持不变，而不会形成布块。我们的算法是基于我们在布匹碰撞的解析中发现的一个特定的顺序，可以用于任何结构建模方法，如弹簧质量或有限元素。本文包括几个真实的涉及无团块的快速运动模拟例子。

关键词：布料动画 碰撞解析 碰撞检测

1 介绍

布块是布料成簇的区域。 在动态布料模拟中，往往会形成布块，妨碍制作逼真动画的过程。由于近年来布料动画系统的稳定性的发展[2,8,12]，模拟快速布料而没有过多的阻尼成为可能。 但是，由于这个问题，用有趣的布料互动来快速移动布料是非常困难的。因此，在本文中，我们提出了一个无块布料碰撞解决方案的新方法，可以实现快速复杂移动布的逼真动画。

快速运动中的布块出现时非常明显，且看起来不自然。当布料节点碰撞速度平均时，会形成布块。由于节点速度是平均的，所以这些节点从此类似地移动，并且看起来像一个团块。虽然人们可能试图用排斥力将布块分成布料片元，或者实现布块厚度，但并不总是可以将布聚集成碎片。尽管接近极限（厚度）限制或排斥力可能有助于布料穿透，但是这种方法本身不能防止布料形成团块，也不能保证布料的穿透自由状态。布料快速运动时，尽管有接近违规的排斥力（惩罚），但它可能会穿透自身，除非严格禁止穿透。如果给予处罚模块的初始布块已经具有自我穿透力，那么即使是严格的布料厚度执行也不能解决自我穿透问题。为了最小化自我穿透的可能，平均布料节点的碰撞速度（暂时将碰撞布料节点作为一个刚性实体来处理）的技术已被广泛使用。然而，这种技术倾向于在碰撞解决后将碰撞的布块留在其附近。

一旦形成了这样的布块，就很难对其进行修正，并且具有这样的块是制造有趣且快速移动的布制动画的障碍。当多个布块同时碰撞时，如图 13 所示的情况 （a，d），纠正自我穿透和解决布块根本不容易处理。

布料碰撞分辨率可以通过如何全面和同时处理多个布料碰撞来分类。 Volino 等人

[16]使用重心节点关系来解决多重碰撞，而 Provot[13]和布里德森等。 文献[6,7]采用冲击区（IZ）的方法，把碰撞布料片元作为一个刚体。由于 Volino 等人提出的方法，保持碰撞节点的位置和速度的重心关系，有效地发现了多个节点的非碰撞位置。但是，如何解决碰撞解决方案导致的后续碰撞是未知的。当布里德森等人。介绍了维持一个确切的布料接近极限 - 一个合理的想法来代表布厚度的想法 - 他们采用了由 Provot 最初提出的 IZ 方法[13]。这种 IZ 方法将同时发生的多个碰撞作为一个组来处理，并且通过平均碰撞的布节点的速度来将参与的碰撞布料片元视为一个刚性实体。从物理上讲，这意味着碰撞解决后碰撞的布料片元表现得像刚体。如果最初的布料片元在快速运动中，平均节点速度的幅度将大大超过来自接近力的排斥速度的幅度，因此会形成块。

1.1 为什么会形成布块

从机械角度来看，布料碰撞是高度无弹性的碰撞，参与的布料碎片失去了大量的动能。 通过这个标记，平均碰撞布节点的速度似乎不会造成任何问题。 但是，这有点误导。 一组离散的布料片元实际上代表了连续的布料片元表面。 当布料碰撞时，由这些布料片元表示的布料片元表面中只有小部分实际上可能发生碰撞。平均所有碰撞布料片元的节点速度意味着假设所有由布料片元表示的布料片元表面处于碰撞 并完成彼此的联系。 这是一个不切实际的假设。在布匹碰撞中假设这种接触的结果是在碰撞分辨率中丢失了不必要的动能。 这是为什么布块会在这样的自我碰撞解决之后形成的物理原因。

1.2 研究意义

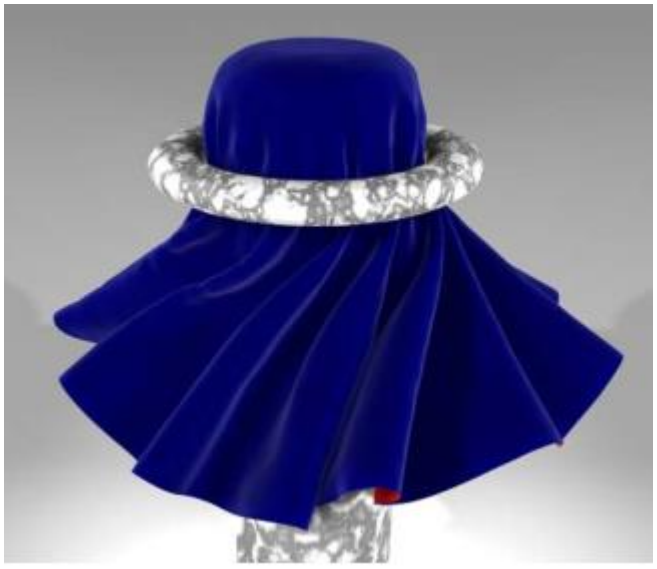

图 1 一个正在快速运动的布料

布料碰撞有两种特殊类型：布料-布料碰撞（自己）和刚体-布料碰撞。 由于这两种布料碰撞代表了物理上不同的情况，所以分别解决它们是合理的。 但是，解决任何一种布料碰撞都可能造成后续的新碰撞。 如果没有处理这种后续冲突的方法，则无法实现

全面完整的冲突解决方案。

本文的第一个贡献是确定布料碰撞响应中可能的顺序，并基于该顺序构建算法。 根据第二部分的观察，我们给出布料碰撞响应中特定的顺序选择的原因。 坚持这一顺序确保碰撞响应的成功，并保证始终保持布料的无穿透状态。

为了避免布料结块，作者采用了前面提出的同时自碰撞解决方法[10]，而它最初是受到文献[1]中提出的刚体同时碰撞解决方法的启发。 我们的自我冲突解决方法使用碰撞集群（CC），并且已经结合了刚体碰撞分辨率。 本文的第二个贡献是提出一种算法，其中综合考虑了两种类型的布料碰撞的分辨率，同时避免了布块。

本文组织如下： 在下一节中，我们确定布料碰撞解决过程中的一个可能的顺序。基于这个讨论，在第 3 节中，我们构建了一个布料碰撞响应算法。 从 3.1 节到 3.5 节，给出了算法的各个阶段。 第 4 节报告了仿真实例的建模选择以及仿真结果。 最后，第 5 节提出我们的结论。

2 布料碰撞的基本问题

在本节中，我们讨论布料碰撞中的几个基本问题，然后再开发我们的算法。 童士业研究了两种布料碰撞（自碰撞与刚体-布料碰撞）和接近违规的相关问题。

2.1 自我碰撞

在讨论布料碰撞解决方案的顺序之前， 首先要解决哪种类型的碰撞问题，研究为什么后续碰撞是解决布料碰撞的重要问题。

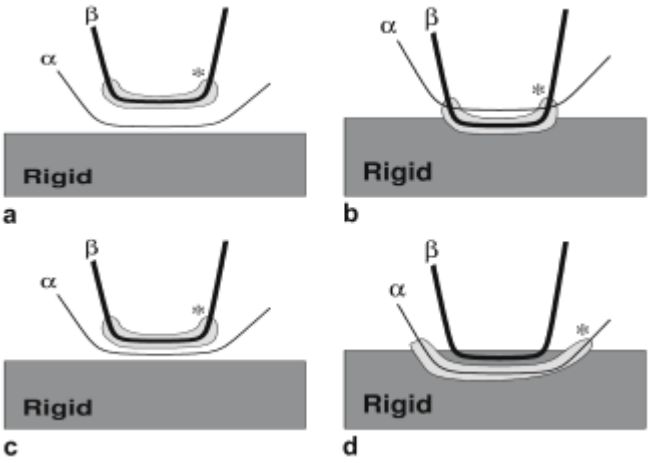

图 2 两个自我碰撞相应的示例

在图 2 中，我们给出了一个例子，其中一个自碰撞无意中解决/创建了一个刚性布料碰撞。我们有一个初始条件，如图 2（a）所示，在这里我们有两个布块和一个阴影刚体在横截面图。如图 2（b）所示，假设两块布料发生碰撞，我们可以得到两种不同的碰撞分辨率结果，如图 2 和图 3 所示。 2（c，d），这取决于我们选择使用的解决方案。如果

我们将 β 相对于 α 移动以避免碰撞，我们将得到与图 2（c）中所示的类似的结果。另一方面，如果相对于 β 移动 α，我们将会有类似于图 2（d）的情况。 （我们的实际自我碰撞解决方法与这些方法不同，它们仅仅是为了解释随后的碰撞而提出的）。在（c）中，我们不知不觉地解决了 β 和刚体之间的碰撞，相反， （d）中创建了刚体。这些新产生的碰撞是碰撞解决后续的碰撞，这表明了为什么人们应该准备处理后续的碰撞，解决布料碰撞，保持连贯的无穿透布料系统。

在布料碰撞中需要注意的另一点是，假设我们忽视了布料对刚性物体的影响，那么由刚性布料碰撞响应引起的任何随后的自身碰撞本质上都是刚性布料碰撞。例如，如果几层布被刚体推动，所有被推动的布块的位置和速度由推动刚性表面决定。图 3 显示了一个情况，即如果刚体向左移动，我们可以预期后续的自碰撞。假设布面（**）和刚体之间有刚性布料碰撞。如果我们通过将碰撞的布料相对于刚性表面移动来解决这些碰撞，我们可能会在（*）中的布料和（**）中的布料之间发生碰撞。考虑到这种情况，我们知道（*）中的布节点位置是相对于（**）中的布块计算的，它们的位置由碰撞的刚性表面确定。这意味着，尽管这些随后的碰撞表现为布料碰撞，但它们是刚性碰撞中固有的。从这个观察结果来看，任何随后的由刚性碰撞分辨率引起的自碰撞都应该与刚性碰撞相结合，布料碰撞决议。由于这个原因，在我们的算法中，刚体分辨率处理被设计成保持相同的相位，直到处理所有后续的碰撞。相反，在自我碰撞反应中，如果任何后来的碰撞是刚性碰撞，则必须调用刚体碰撞解决模块。因此，在我们的算法中，首先解决自碰撞，然后才开始刚体碰撞解析。图 12 和图 8 分别示出了刚性布料和自身碰撞分辨率的程序，并且在第 3 节中提出了整个布料碰撞响应算法。

2.2 接近违规限制

PV 约束（实施厚度）已被广泛用于避免实际穿透。 防止布块彼此过于靠近也被认为能有效地减少来自数值精度误差的错误碰撞检测。

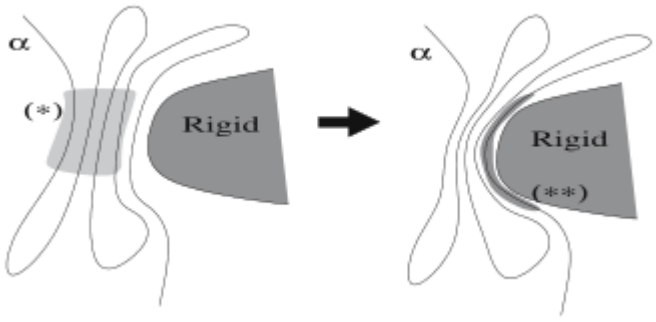

图 3：布料层与刚体之间的碰撞

然而，当 PV 被穿透时，处理 PV 会加剧目前的穿透。 当一组布料片元被给予邻近违规处理模块时，布料片元预计不会被穿透。 换句话说，给予接近违规处理模块的布必须是无穿透的。 否则，具有穿透性的 PV 不能被正确处理。 图 4 显示了两个穿透面，

而它们的节点处于接近违规状态。 如果在这种情况下处理 PV，则节点 N1，N2 和 N3 将从另一个三角形所在的表面移开。 显然这只会加剧事先的穿透。 为了避免这种情况，我们的算法在应用接近违规约束之前解决了所有的穿透问题。

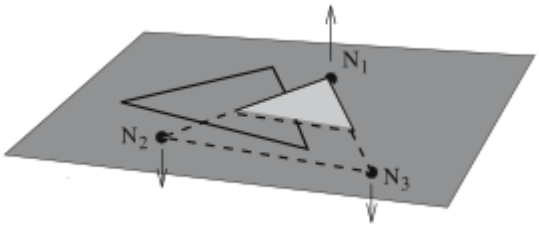

图 4 一个穿透示例

3 算法

根据到目前为止所讨论的问题，我们得出以下结论：

1.自我碰撞要在刚体-布料碰撞之前解决。

2.在 PVs 之前要先穿透。

因此，我们选择的特定冲突解决顺序是 self=>self PV=>rigid-cloth =>rigid-cloth PV.

使用这个顺序，图 5 中所示的算法被建立，其中初始织物状态被要求是无穿透的。只要最初的布料无穿透，我们算法的最终结果就是保证无穿透。 在每个模块中，通过不让任何穿透通过未解决的下一个时间步骤，我们保持系统的有效性。

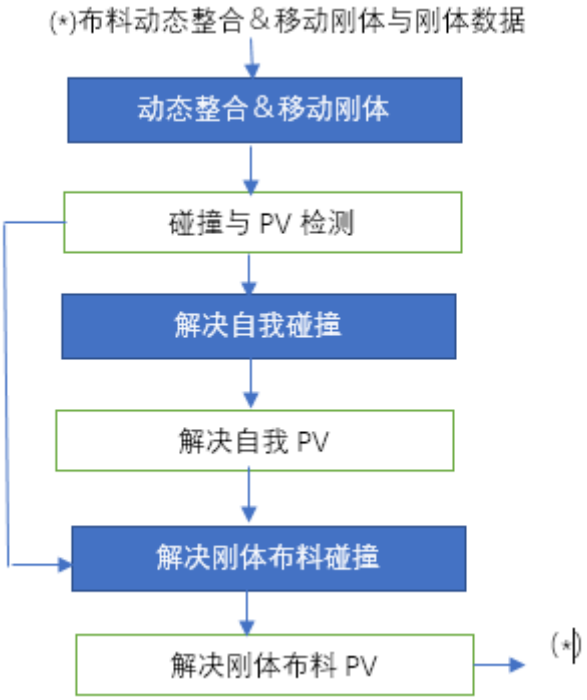

图 5 布料碰撞相应算法

注意碰撞和 PV 的检测仅在动态积分之后执行一次。然而，此后的每个分解阶段可能产生在检测时不存在的条件。 因此，在进入下一个模块之前，必须检查并解决来自每个阶段的所有修改节点的碰撞，以确保系统的正确性。 在下面的章节中，我们的算法的每一个阶段都有详细的解释。

3.1 碰撞与 PV 的检测

在本节中，我们将讨论我们算法的第一阶段 - 碰撞检测。 对所有可能的边缘对和面对节点对执行冲突检测，而所有的边，节点和面被注册在轴对齐的体素中。 在任何瞬时的模拟中，一个元素可能同时包含在几个体素中[17]。 每个体素是立方体，其中一个体素侧的长度比最长边长稍长。 对三维像素中的所有元素对进行可能的碰撞或违规测试。 在碰撞检测的情况下，在前一时间步中占据体素的元素和从前一时间步到当前时间步通过体素的元素也要进行碰撞测试。

碰撞数据不是只报告碰撞对，也为了我们的同时碰撞解决的目的而存储。 下面解释所需的数据结构。

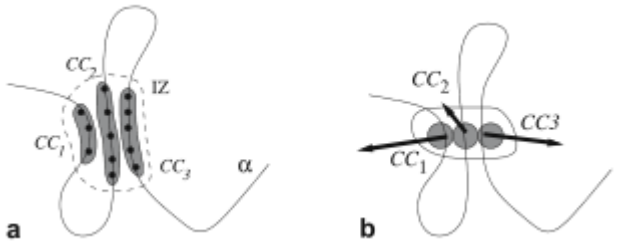

图 6 a 与 b 保存碰撞与 PV 数据，a 用于在 IZ 中识别 CC，b 碰撞相应的单位

在找到所有的碰撞对之后，我们将所有相互碰撞的实体（面，节点和边）组合成一个 IZ。 IZ 是同时代表多个碰撞中的一组布块的数据结构。在图 6（a）中，虚线区域 IZ 表示具有三块布块 CC_1，CC_2 和 CC_3 的布块。如果一个边是 IZ 的一个成员，那么它就是同一个 IZ 的两个结束点代表成员，同样的规则也适用于成员面的结点。在 IZ 中，节点根据边缘连接性进行分组。这种边缘连接的节点被称为碰撞簇（CC）。可以在文献[10]中找到对 CC 和 IZ 更严格的定义。图 6（a）显示衣物的横截面图，圆圈表示 IZ，阴影区域为 CC。如图 9 所示，这些 CC 将作为碰撞响应的单元进行处理。将这些 CC 作为碰撞单位来处理，而不是对单个碰撞作出反应，使我们能够同时解决碰撞，同时减少对布块的担忧。

3.1.1 虚假碰撞检测

为了检测在一个时间步骤内发生的碰撞，我们使用从前一个时间步到当前时间步的节点位置的轨迹，同时使用[13]中提出的碰撞检测方法。假定每个节点的轨迹是线性的，并且节点的速度是恒定的。 但是，这种检测方案有时会给我们提供虚假的碰撞报告。

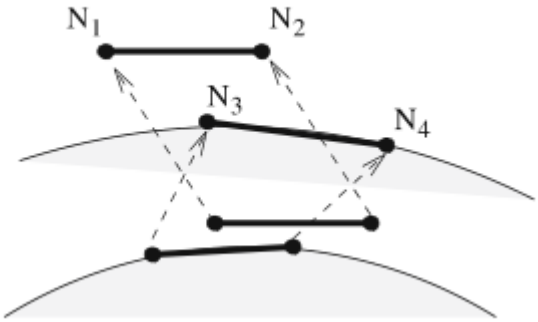

图 7 虚假碰撞判定

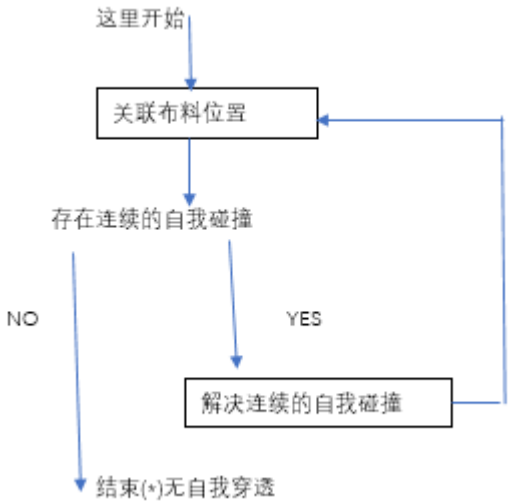

图 8 一个自我碰撞解决方案模型

例如， 图 7 示出布节点 N1 和 N2 以及硬身体节点 N3 和 N4 在虚线方向上移动的情况。 显然，这不是一个碰撞的情况。 然而，由于我们假定了速度不变的线性轨迹，因此可以将其报告为碰撞。 在一个时间步内可能有一个时间 t，当所有的节点是共面的，并且由这些节点构成的边相交。 在这种情况下，碰撞检测模块报告碰撞。 同样，我们也可以对一个人脸节点进行错误的碰撞检测。当需要精确的碰撞检测时，可以通过检查碰撞后碰撞块的法向是否翻转节点方向来消除错误的碰撞。 对于所提供的示例，检查相对方位将过滤错误的碰撞检测。 在我们的测试模拟中，翻转相对方向的检查用于刚性布料碰撞检测。

3.1.2 PV 的检测

仅在当前时间步的节点位置上测试接近违规。 使用节点面和边缘对来测试 PV，看它们是否违反了接近约束（在我们的动画中为 2mm）。 在检测到 PV 时，检测模块将 PV 实体对（节点面，边缘）存储到每个节点。 存储在每个节点 PV 中的这些 PV 违例信息列表稍后会在 PV 分辨率模块中使用，详见 3.3 节和 3.5 节。

3.2 自我碰撞反应

给定检测到的碰撞信息，我们首先解决自身碰撞。 在本节中，我们解释如何使用同步方法来解决包括多个自身碰撞的自身碰撞。 本节还介绍了 CC 方法与 IZ 方法的根本区别，并说明了为什么它适用于快速移动的布料。

3.2.1 多次自碰撞的解决方法

据报道，解决多次自相撞是许多研究小组的难题。 针对这个问题已经提出了几种方法，包括[6,13]的 IZ 方法和[10]的 CC 方法。 IZ 分辨率方法对所有碰撞粒子的速度进行了加速[13]，并利用初始条件计算出的角速度进一步调整每个粒子的速度，以保留旋转信息[6]。

图 9 显示了多个自身碰撞，其中 A，B 和 C 表示 CC。 CC 的初始速度分别为 v A，v_B 和 v_C，其中 $v_B = 0$。当 $v_{A,x}$ 和 $v_{A,y}$ 代表 v_A 的 x 轴和 y 轴分量时，令 $v_{A,y} = -v_{C,y}$，$v_{A,x} = -v_{C,x}$。

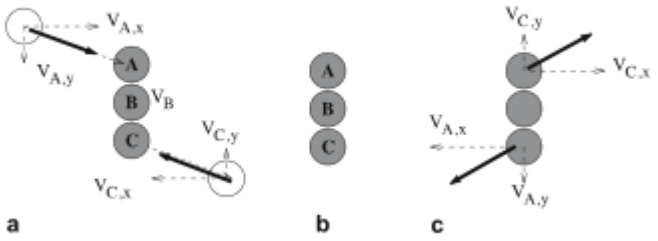

图 9 三个碰撞中的 CCS

考虑到这种情况，IZ 方法给出了图 9（b）所示的结果，其中所有的球都具有零速度。由于球 A 和 B 在碰撞中失去了所有的动力，所以它们将保持在另一个球的附近。因此球似乎彼此粘附，这是不可取的。

图 9（c）显示了 CC 方法的结果。CC 方法使用同时碰撞响应方法。 每个球（CC）的处理就好像它是一个独立的实体。 因此，在我们的例子中，A，B 和 C 代表所有不同的 CC。 CC 方法将碰撞视为弹性碰撞[4]，将其快速分开。 因此，解析结果与三个弹性球的情况相同，其结果保存了初始动量，并且比未使用 CC 的情况下的动能损失小。因此，布节点不会有任何不希望的粘着效应。

然而，由于时间和空间上的一致性，当多个布节点发生碰撞时，它们附近的节点往往也会在很短的时间内发生碰撞。这就意味着在 CC 方法中，碰撞区域的速度可以保持交换。 我们的算法通过使用 IZ 结壳改善了这个问题，正如 3.3 节所讨论的那样。 我们的方法来解决这个问题是可能的，因为我们的算法分别和不同地处理自我 PV 和自我碰撞。

3.2.2 应用同步方法

在本节中，我们讨论如何执行自碰撞的实际解决方案。 解决自身碰撞就是寻找适当

的碰撞节点的速度和位置。节点的合适位置是节点本身以及连接到该节点的面和边与环境中的任何物体不发生碰撞的位置。 为一组节点找到这样的位置可能是一项艰巨的任务。 然而，我们的算法利用了这样一个事实，即在前一个时间步的末尾，这些节点的位置是无碰撞的。 通常使用的时间步长（10-0.1 毫秒），即使在节点处于快速运动的情况下，节点在一个时间步长内也只移动很小的距离。为此，我们使用前一个时间步的节点位置作为碰撞节点的当前位置。 当布节点发生自身碰撞时，我们只需将上一个时间步的节点位置恢复为当前位置。

为了在一个 IZ 中找到所有 CC 的速度，我们首先必须确定 CC 的碰撞方向。 在我们的方法中，碰撞方向是所有 CC 法线的平均值，其中 CC 法线是 CC 中所有节点法线的平均值。 速度调整仅对节点速度平行于该碰撞方向上的组件来执行。。 碰撞节点在自我碰撞后保持其速度的切向分量。

令 n 为 CC 的个数，v_i 为碰撞前第 i 个 CC，CC_i 的速度，v_i'为 CC_i 的碰撞响应后的速度。 给定 v_i，我们必须为所有 CC 找到 v_i'。根据牛顿物理学，我们得到

$$(v_j' - v_i') = k_e(v_i - v_j) \qquad (1)$$

对于所有碰撞对 CC_i 和 CC j，其中 k_e 表示弹性系数。 根据动量守恒定律，我们有 $\sum_i m_i\, v_i = \sum_i m_i\, v_i'$，其中 m_i 是 CC_i 的质量。 值得注意的是，由于我们假设所有布节点具有相同的质量，m_i 就是 CC_i 中的节点数目。 在弹性碰撞的情况下，我们有

$$v_i' = total_{2v} - v_i v_i \qquad (2)$$

其中$total_{2v} = (\sum_i 2m_i v_i)/(\sum_i m_i)$

循环碰撞的情况是我们在碰撞中找到一个循环的情况。 例如，当 A 和 B 以及 B 和 C 之间存在碰撞时，如果 C 和 A 也发生碰撞，则称为循环碰撞情况。 循环碰撞也可以通过公式 2，因为公式 1 在循环碰撞和单个自身碰撞的情况下仍然适用于所有的碰撞对。

虽然这些恢复的节点都没有任何内部的冲突，但是由于附近节点的位置已经改变，所以我们必须检查后续的冲突。 在随后的碰撞中，我们可以恢复相关节点上一个时间步的位置。 需要考虑的一个有趣的地方就是在后续碰撞中遇到另一个 IZ 的情况。 但是，由于我们只是恢复前一个时间步的位置，所以当前的解决方案不会影响到其他 IZ 的任何解决过程。 此外，由于前一时间步的所有节点位置都没有碰撞，所以布节点的数量与前一时间步不变，因此自冲突解决过程在计算上是有界的，并且保证了工作。

3.3 自我 PV 解决方案

自我 PV 解决模块的目标是尽可能减少 PV。 自 PV 分辨率不会产生任何后续自碰撞，因为 PV 解决方案只包含碰撞对节点速度的影响。

许多研究人员通过在布料和布料之间施加小的弹簧力来防止渗透[2, 5, 4]，已经采用了接近限制。 尽管接近性约束是一种可行的技术，但由于易于理解和使用简单，所以它有时可以穿透。 布里德森等[6]通过模拟布厚度提出了更严格的接近约束。 他们的方法

要求布块保持精确的布厚度，以减少实际的碰撞。 然而，在他们的方法中，由于布点节点的速度在积分时间步骤结束时被调整以达到接近极限，所以布点在 PV 解析过程中往往会失去大量的动量。

有接近约束的另一个重要的一点是，可以通过不允许两个实体过于接近或穿透来避免在检测碰撞时发生数值错误的可能性。 多个研究小组使用大多数定向方法修补了布料穿透[3,15]。 但是，在多重碰撞的情况下，这种多数取向数据并不总是可靠的。 此外，在处理 PV 时穿透可能会加剧穿透，如第 2.2 节所述。 因此，在我们的方法中，在处理 PV 之前确保无穿透的状况，以确保系统的完整性。

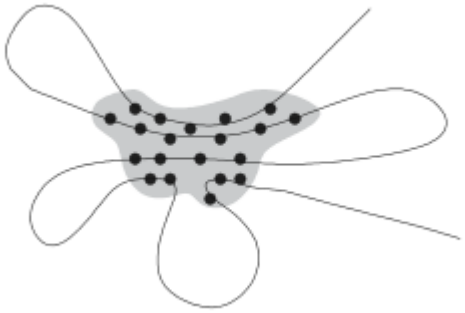

图 10 多个邻近违规（PV）中的一组节点

图 10 表示一块布料片元的横截面图，其中阴影区域表示多个 PV 中的布节点。 为了解决 PV,我们首先在 PV 违例中的面和边的列表中识别关于节点的最接近的面或边。从节点到最近实体的距离用于计算接近约束力。 一旦我们确定了最接近的实体，我们就会按照缓解最严重的 PV 违规的方向向节点施加接近力。

3.4 刚体-布料碰撞响应

在刚体-布料碰撞响应中，仅在布节点上进行解析，以发现相对于刚体的非碰撞位置和适当的速度，而碰撞的刚性节点是完整的。 给刚体-布料碰撞响应模块的输入没有自身碰撞，刚性解决方案的结果应该没有刚体-布料和自我碰撞。检测到的刚体-布料布碰撞是以一组形式给出的 的 IZ。 然后解决方案分别解析每个 IZ。

刚体-布料碰撞解决方案从识别相互碰撞布节点的相关刚性面开始。 布节点的相关刚性面是一组连接到碰撞刚体的面。 例如，当布料 n_c 是与刚性节点 n_r 碰撞的面 f_c 中的节点之一时，刚性节点 n_r 的相邻面被包括在布节点 n_c 的相关刚性面中。 在识别出这种相关的刚性面之后，我们从布节点 n_c 中识别出最接近的刚性面 f_*。 这个刚性面 f_*在节点 n_c 的刚体解决方案中起着关键作用。 对于解决方案，布节点 n_c 相对于前一时刻的刚性面 f_*的位置的相对位置被恢复，给出刚性面 f_*的当前位置。

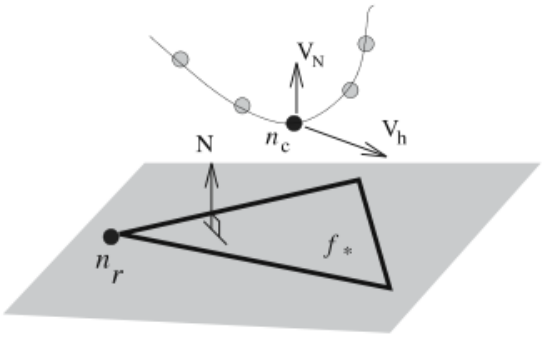

图 11 刚体面 f*与布料节点 nc 的碰撞

n_c 的速度基于面法线 N 和 f_*的速度 v_r 来解析。 如图 11 所示，当 v 是碰撞前 n_c 的速度，v'是 v 之后的一个，v_N 是 v 的法向分量，v_h 是 v 的切向分量，刚体-布料碰撞后只有法向 速度的成分受到影响。V_n'的法向速度有以下公式计算：

$$v'_N = \begin{cases} k_v v_N & if\ v_n - v_N^r \geq 0 \\ v_N^r & otherwise \end{cases} \qquad (3)$$

其中 k_v 是冲击系数，$0 \geqslant k_v \geqslant 1$。当布节点 n c 远离刚性表面 f_*，即 $v_N - v_N^r \geqslant 0$ 时，n_c 的法向速度是完整的。 否则布节点将继承刚性面的法向速度。 这个解决方案是基于这样的直觉：刚体-布料碰撞会影响碰撞时碰撞布节点的速度。

v_h'的切向分量的摩擦力如下方程确定：

$$f_t = k_f |v'_N - v_N|(v_h - v_h^r) \qquad (4)$$

其中 k_f 是摩擦系数。 再次，当布节点 n_c 从刚性表面移开时，它是$|v'_N - v_N| = 0$.因此节点在这种情况下没有摩擦。

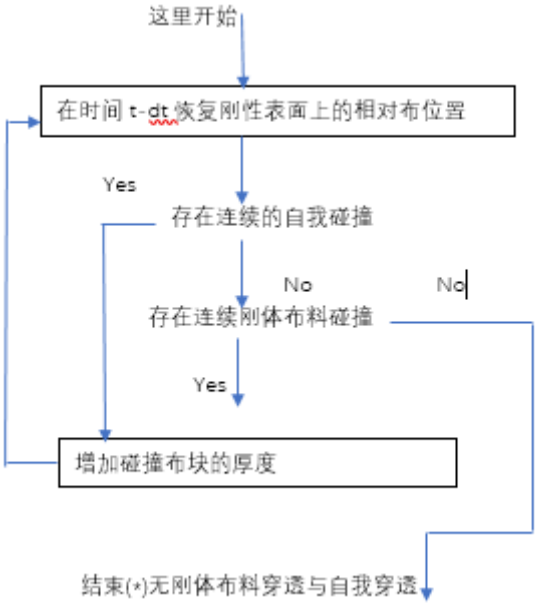

图 12：刚体布料碰撞解决方案模型

一旦节点位置解决了，我们检查它们是否产生了新的碰撞，如图 12 所示。我们不仅要检查解析的节点，还要检查连接这些节点的边和面，因为这些元素也受位置 节点

的变化。 解决的布料片元可能会发现造成其他自我碰撞。 如果解决的布料片元与另一块布料片元相交，新的碰撞的布料片元处于刚体-布料碰撞的情况。 新布料片元中的节点的解析位置也以相同的方式计算。 重复这个迭代直到我们没有发现新的自身碰撞。在这种迭代式刚性分辨率失效的情况下，虽然很少发生，但我们使用刚体表面的重心节点关系。 计算新的布节点位置以恢复前一时间步的布节点相对于刚体的相对位置

3.5 刚体-碰撞 PV 解决方案

刚体-碰撞 PV 解决方案与自身 PV 解决方案类似，除了布节点的速度将与自身 PV 响应的情况类似地进行调节。

4 实验

对于本文提出的例子，基于粒子的系统与三角形布面一起使用。 对于弯曲力，为每个非边界边添加一个交叉边，如图 14 所示。采用文献[2]提出的隐式共轭梯度法进行积分，采用文献[8]提出的屈曲处理。 然而，我们的算法是独立于布料结构的任何具体定义和动态模拟方法的选择。

在我们的例子中，使用由 Bridson 等人提出的碰撞避免方法[6]，通过环细分方法[9,11]将布表面细分两次。我们的例子中的环境在细分后有大约 10 万人的面孔。在 Linux 机器上使用 3GHz Pentium IV CPU 的环境下，取决于当前涉及的碰撞次数，通常的帧时间是 1-4 分钟。

图 13 快速运动布料示例

图 13 所示的照片是来自模拟的快照。 图 13（a，c）中的布料显示了多次布料碰撞的情况。 圆圈区域是以前的方法可能由于广泛的自我碰撞和刚体-布料碰撞相互作用而失败的地方。 图 13（b，d）示出了图 1 和 2 中的框架的从下面看的视图。 3（a，c），中间的圆柱体不可见。 图 13（f，e）中所示的舞者的图像是使用我们的碰撞解析模块从动画中获得的快照。 （e）中的布比（f）中的布更硬，并且以（f）中的跳动速度的两倍来模拟。

5 讨论

本文的主要贡献是确定处理布料碰撞的特定顺序，并根据此顺序构建布料碰撞响应算法。 我们的算法综合考虑两种布料碰撞，并结合适当的迭代后续碰撞问题处理，这对于提供可靠的布料碰撞响应算法是必不可少的。 我们的算法使我们能够制作出无布块且快速移动的布料运动。

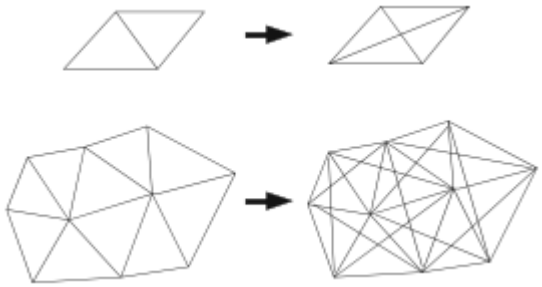

图 14 向三角形网格里增加边

致谢：作者由 NSF ITR 0205671 发表。 1 和 13（a，b，c，d）由 Paul Kanyuk 渲染。我们感谢他的工作。 作者还感谢 AUTODESK Maya 所提供的服装设计支持。

参考文献

1. Baraff, D.: Analytical methods for dynamic simulation of non-penetrating rigid bodies.In Proceedings of SIGGRAPH 23(3),Computer Graphics Proceedings, Annual Conference Series, pp. 223–232. ACM Press/ACM SIGGRAPH (1989)
2. Baraff, D., Witkin, A.: Large steps in cloth simulation. In Proceedings of SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, pp. 43–53. ACM Press/ACM SIGGRAPH (1998)
3. Baraff, D., Witkin, A., Kass, M.: Untangling cloth. In Proceedings of SIGGRAPH, Vol. 22(3) of Computer Graphics Proceedings, Annual Conference Series, pp. 862–870. ACM Press/ACM SIGGRAPH (2003)
4. Beer, F.P., Johnston-Jr., E.R.: Vector Mechanics for Engineers – Dynamics.WCB/McGraw-Hill
5. Breen, D.E., House, D.H., Wozny, M.J.: Predicting the drape of woven cloth using

interacting particles. In Proceedings of SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, pp. 365–372. ACM Press/ACM SIGGRAPH (1994)

6. Bridson, R., Fedkiw, R., Anderson, J.: Robust treatment of collisions, contact and friction for cloth animation. In Proceedings of SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series,pp. 594–603. ACM Press/ACM SIGGRAPH (2002)
7. Bridson, R., Marino, S., Fedkiw, R.: Simulation of clothing with folds and wrinkles. In Symposium on Computer Animation, pp. 28–36. Eurographics/ACM Siggraph, Eurographics Association (2003)
8. Choi, K.-J., Ko, H.-S.: Stable but responsive cloth. In Proceedings of SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series,pp. 604–611. ACM Press/ACM SIGGRAPH (2002)
9. deRose, T., Kass, M., Truong, T.: Subdivision surfaces in character animation. In Proceedings of SIGGRAPH,Computer Graphics Proceedings, Annual Conference Series, pp. 85–94. ACM Press/ACM SIGGRAPH (1998)
10. Huh, S., Metaxas, D.N., Badler, N.I.: Collision resolutions in cloth simulation. In IEEE Proceedings of Computer Animation, Seoul, South Korea (2001)
11. Loop, C.: Triangle mesh subdivision with bounded curvature and the convex hull property. Technical Report MSR-TR-2001-24. Microsoft Research (1995)
12. Provot, X.: Deformation constraints in a mass-sprint model to describe rigid cloth behavior. In Graphics Interface, pp. 147-155. Graphics Interface (1995)
13. Provot, X.: Collision and self-collision handling in cloth model dedicated to design garments. In Proc. of Graphics Interface, pp. 177–189. Graphics Interface (1997)
14. Volino, P., Courchesne, M., Thalmann,N.M.: Versatile and efficient techniques forsimulating cloth and other deformable objects. In Proceedings of SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, pp. 137–144. ACM Press/ACM SIGGRAPH (1995)
15. Volino, P., Magnenat-Thalmann, N.:Collision and self-collision detection:efficient and robust solutions for highly deformable surfaces. In: Terzopoulos, D.,Thalmann, D. (eds.) Computer Animation and Simulation ’95, pp. 55–65. Springer,Berlin Heidelberg New York (1995)
16. Volino, P., Magnenat-Thalmann, N.:Accurate collision response on polygonal meshes. In Proceedings of the 2000 Conference on Computer Animation, pp. 154–163. IEEE Computer Society (2000)
17. Zhang, D., Yuen, M.: Collision detection for clothed human animation. In Proceedings of the 8th Pacific Graphics Conference on Computer Graphics and Application, pp. 328–337. IEEE Computer Society (2000)

本文译自：Suejung B. Huh,Dimitris N. Metaxas, et al．A collision resolution algorithm for clump-free fast moving cloth：Visual Comput (2006) 22: 434–444. Springer-Verlag New York, Inc. 2006