

文章编号: 2095-2163(2022)02-0001-05

中图分类号: TP 391.9

文献标志码: A

基于空间线性投影的连续碰撞检测算法

万燕, 符亚玲, 姚砺

(东华大学 计算机科学与技术学院, 上海 201620)

摘要: 为提高复杂的高精度模型在物理仿真中的连续碰撞检测效率, 本文提出了一种基于空间线性投影的连续碰撞检测算法。算法采用两级碰撞算法框架, 在高层剔除阶段, 综合考虑构造难度与剔除效果, 确定使用离散导向多面体包围盒(26-DOPs)对基本图元进行第一次剔除; 在低层剔除阶段, 针对非穿透性滤波器(DNF)只能剔除不满足共面条件的碰撞对这一弊端, 提出空间线性投影滤波器(Spatial Linear Projection Filter, SLPF)对不满足内部条件的碰撞对进行剔除; 综合使用这两个滤波器, 对基本图元进行第二次剔除。在funnel数据集和cloth_ball数据集上进行实验, 相比于仅使用非穿透性滤波器, 结合使用非穿透性滤波器和空间线性投影滤波器时, 点面碰撞剔除效率分别提高了6.71%和20.11%, 边边碰撞剔除效率分别提高了5.37%和15.98%。

关键词: 连续碰撞检测; 滤波器; 高层剔除; 低层剔除

Continuous collision detection algorithm based on spatial linear projection

WAN Yan, FU Yaling, YAO Li

(College of Computer and Technology, Donghua University, Shanghai 201620, China)

[Abstract] In order to improve the efficiency of continuous collision detection for complex and high precision models in physical simulation, a continuous collision detection algorithm based on spatial linear projection is proposed. The two-phase collision algorithm framework is used in our algorithm. In the high-level culling phase, considering the difficulty of construction and the effect of elimination, the discrete oriented polyhedron bounding box (26-DOPs) is determined to be used for the first elimination of basic elements. In the low-level culling phase, in view of the disadvantage that the non-penetrating filters (DNF) can only cull the collision pairs that do not meet the coplanar condition, the spatial linear projection filter (SLPF) is proposed to cull the collision pairs that do not meet the internal condition. Combined with these two filters, the basic elements are eliminated for the second time. Experiments on the funnel dataset and cloth_ball dataset show that compared with DNF alone, when DNF and SLPF are used, the culling efficiency of vertex face (VF) collision increase by 6.71% and 20.11%, and that of edge edge (EE) collision is increases by 5.37% and 15.98%.

[Key words] continuous collision detection; filter; high level culling; low level culling

0 引言

随着计算机仿真的发展, 碰撞检测作为其中最重要的环节之一, 也受到了计算机图形学研究人员的极大关注^[1]。近年来, 研究人员在碰撞检测领域提出了各种有效的算法, 并开发了相应的软件包, 但是由于复杂的高精度模型往往包含数以万计的三角面, 要在短时间内检测碰撞信息是一项极具挑战性的工作。

目前, 大部分实时性较好的碰撞检测算法都是传统的离散碰撞检测算法, 只在离散的时间点上对目标进行碰撞检测, 这类算法可能会造成碰撞缺失, 因此形成隧道效应, 不仅使得场景缺乏真实感, 而且

会有过大的计算开销^[2]。有研究者提出一种连续碰撞算法, 在两个离散时间点之间使用线性插值的方法来描述物体运动轨迹, 检查两个时间点间是否发生碰撞, 同时计算首次碰撞时间^[3]。碰撞丢失等让场景产生不真实感的问题在仿真系统中需要尽量避免, 因此必须采用连续碰撞检测而不是离散碰撞检测以避免出现假象。

在碰撞检测中, 碰撞目标由大量三角基元构成, 一对三角对的碰撞检测可转化为9对点面(VF)基本碰撞测试和6对边边(EE)基本碰撞测试, 这15对基本测试最终转换为求解三阶代数方程的问题^[4]。求解三阶代数方程代价较大, 因此对碰撞基元进行快速剔除, 避免后续的求解方程是很有必要

作者简介: 万燕(1970-), 女, 博士, 教授, 主要研究方向: 图像处理和机器学习; 符亚玲(1996-), 女, 硕士研究生, 主要研究方向: 计算机图形学; 姚砺(1967-), 男, 博士, 副教授, 主要研究方向: 图像处理。

通讯作者: 姚砺 Email: yaoli@dhu.edu.cn

收稿日期: 2021-10-19

的,目前普遍采用两级碰撞算法对三角对进行剔除,即高层剔除阶段和低层剔除阶段,大大减少求解方程的数量。经过两层剔除后,再对剩下的相对较少的基本图元对进行精确碰撞检测^[5]。由于精确碰撞检测花费时间较多,因此尽量剔除不可能相交的基元,提高整体的碰撞检测效率。

两级碰撞检测算法仅仅规定了碰撞对需经过高层剔除与低层剔除,再执行精确碰撞检测,而两层剔除各自使用何种算法未作设定,由研究者自行设计。目前存在的低层剔除算法要么计算量大或者构造难度高,要么剔除率不高,因此,本文提出了空间线性

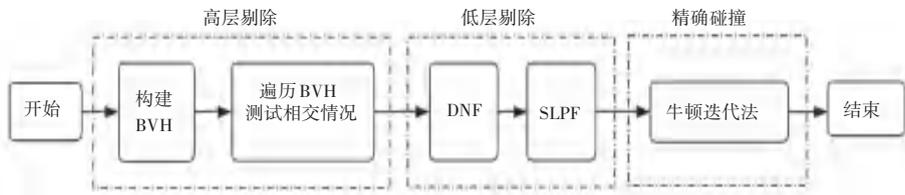


图1 整体算法框架图

Fig. 1 Flow chart of the proposed algorithm

1.1 高层剔除

高层剔除阶段主要是对场景中明显不相交的基元进行整体剔除,通常使用包围盒层次方法(BVH)或者空间分割法。包围盒的种类很多,如球形包围盒、AABB包围盒、OBB包围盒和kDOPs包围盒;空间分割常用的有均匀网格、二维空间分割树(BSP)和八叉树(Octree)。

本文的高层剔除选择包围盒层次结构(BVH),以包围盒(BV)作为结点的树,根结点是一个包括所有物体的包围盒,叶子结点是最小的包围盒,通常只包含一个图元,一个简单的包围盒层次结构(二叉BVH树)如图2所示。不同的包围盒,其构成的BVH剔除效果不同,在综合考虑剔除效果与时间效率后,本文选用k-DOPs包围盒。 k 值不同时其剔除效果也不相同, k 的取值有6、8、12、14、18和26,理论上 k 值越大剔除效果越好,树的构建和更新所耗费的时间越多,但相交测试的时间越少。一般情况下,相交测试时间远大于构建树的时间,因此选定 k 值为26。此外,为了进一步提升包围盒剔除效果,本文在传统BVH上增加了一层额外包围盒,使得包围盒最小包围的是三角形的点和边,而不是三角形本身。

1.2 低层剔除

经过高层剔除,仍然有很多冗余三角面片不能被剔除,低层剔除将待检测图元对的碰撞检测转换为点面测试和边边测试,每个基本测试又可以分为

投影滤波器(SLPF),与文献[6]中的非穿透性过滤器(DNF)相结合对基本图元进行低层剔除,可以有效提高剔除效率。

1 研究方法

本文采用两级碰撞检测算法对三角对进行剔除,快速减少大量冗余三角对。高层剔除选择包围盒层次结构(BVH),非穿透性过滤器(DNF)与本文提出的空间线性投影滤波器(SLPF)结合作为低层剔除方法。整体算法框架如图1所示。

两个部分:共面测试和内部测试。

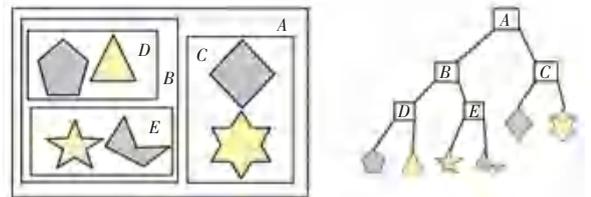


图2 简单场景二叉BVH树

Fig. 2 Simple example of BVH

以点面测试(测试点面对的碰撞情况)为例,若点和面发生碰撞,则点 P 在 $\triangle ABC$ 内部,即 P 、 A 、 B 、 C 4点共面(共面条件);同时碰撞点 P 在 $\triangle ABC$ 内(内部条件),如图3所示。判断碰撞对是否满足共面条件称为共面测试,判断碰撞对是否满足内部条件的称为内部测试。

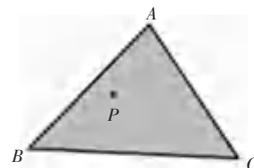


图3 点对面发生碰撞时的位置关系

Fig. 3 Position of VF-pair when they collide

低层剔除使用的有代表三角形、孤集和过滤器等方法。

本文的低层剔除采用两个过滤器:非穿透性过滤器(DNF)和本文提出的空间线性投影滤波器(SLPF)。由于碰撞对确定碰撞需要满足两个条件:

共面条件和内部条件。DNF 可以剔除不满足共面条件的碰撞对,而本文提出的 SLPF 可以剔除部分不满足内部条件的碰撞对,因此综合使用这两个滤波器,可以剔除大量不满足碰撞条件的碰撞对,得到效果较好的低层剔除。在多个模型上实验证明,结合使用 DNF 和 SLPF 的方法比仅使用 DNF 方法剔除数量更多。

2 空间线性投影滤波器

空间线性投影滤波器 (SLPF) 是将碰撞对投影到空间中的直线上,根据直线上投影点的位置关系,直观简单地判断碰撞对原来的位置关系,可以剔除大部分不满足内部条件的碰撞对。

2.1 符号说明

V_0, V_1 和 V_t 分别表示点 V 在 $t = 0, t = 1$, 以及任意 $t \in [0, 1]$ 的情况。

P, A, B, C 表示点面对 ($VF - pair$) 的运动点和三角形的 3 个顶点, A, B, C, D 表示边边对 ($EE - pair$) 的两条边 AB, CD 的 4 个顶点。

\times, \cdot 和 $*$ 分别表示向量叉乘、点乘以及普通乘法。

黑斜体表示向量,例如: $s, AtPt, atpt$ 。

2.2 空间线性投影滤波器的理论介绍

SLPF 是将空间中的碰撞对投影到空间中的某条直线上,根据投影点的位置关系来剔除不可能发生碰撞的碰撞对。在点面碰撞中,若点的投影点始终在三角形 3 个顶点的投影点的同一侧,则可剔除;在边边碰撞中,若一条边的两个顶点的投影点始终在另一条边的两个顶点的投影点的同一侧,则可剔除。

定理 1 在任意时间间隔内,对于特定的投影空间,如果两个变形基元在投影空间中没有相交,则这两个基元在其原空间中也不会相交。

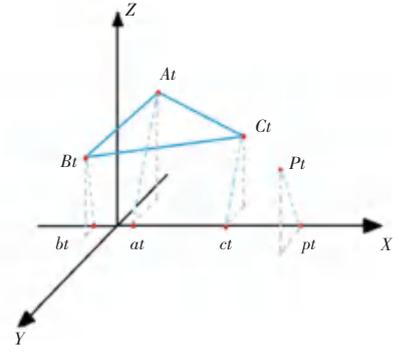
证明 假设在原本的空间中两个变形基元相交,那么在投影空间中这两个基元也一定相交,与两个基元在投影空间不相交矛盾,故假设不成立,定理 1 成立。

下面分点面对 ($VF - pair$) 和边边对 ($EE - pair$) 两种情况分别说明 SLPF 的原理。

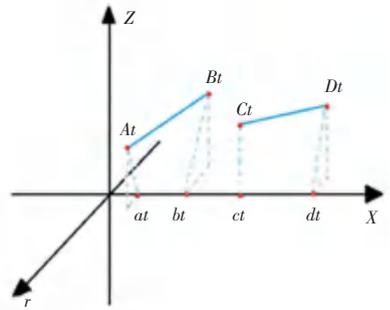
(1) 点面对: 将 $VF - pair$ 的 4 个点 P_t, A_t, B_t, C_t 投影到某条直线上, 记为 pt, at, bt, ct 。由定理 1 可知,若整个时间间隔内没有相交,那么必定存在一条直线,有 pt 始终在 at, bt, ct 的左边 (或右边), 则这一对点面一定不发生碰撞,反之则不一定发生碰撞。

点面对在方向向量为 $(1, 0, 0)$ 即 X 轴上的投影情况, 如图 4(a) 所示。

(2) 边边对: 将 $EE - pair$ 的 4 个点 A_t, B_t, C_t, D_t 投影到某条直线上, 记为 at, bt, ct, dt 。由定理 1 可知,若整个时间间隔内没有相交,那么必定存在一条直线,有 at, bt 始终在 ct, dt 的左边 (或右边), 则这一对边边一定不发生碰撞,反之则不一定发生碰撞。边边对在方向向量为 $(1, 0, 0)$ 即 X 轴上的投影情况, 如图 4(b) 所示。



(a) 点面对在 X 轴上投影



(b) 边边对在 X 轴上投影

图 4 碰撞对在 X 轴上投影

Fig. 4 The projection of VF-pair and EE-pair onto the X-axis

2.3 空间线性投影滤波器计算推导

为将位置关系转换为能够用程序计算的公式, 下面进行计算推导。

记投影向量的单位向量为 s , 以点 A_t 和 P_t 为例。如果 P_t 的投影点一直在 A_t 的投影点的右侧, 则有线段 A_tP_t 构成的向量 $AtPt$ 在以 s 为单位向量的直线上的投影为 $atpt$ 与 s 的点乘大于 0, 即 $atpt \cdot s > 0$ 一直成立, 如果 P_t 的投影点一直在 A_t 的投影点的左侧, 则 $atpt \cdot s < 0$ 一直成立。

$$atpt \cdot s = (Opt - Oat) \cdot s = |Opt| * \cos < Opt, s > * s \cdot s - |Oat| * \cos < Oat, s > * s \cdot s = |Opt| * \cos < Opt, s > * s \cdot s - |Oat| * \cos < Oat, s > * s \cdot s = |Opt| * \frac{Opt \cdot s}{|Opt| * |s|} * s \cdot s -$$

$$|OAt| * \frac{OAt \cdot s}{|OPt| * |s|} * s \cdot s = OPt \cdot s * s \cdot s - OAt \cdot s * s \cdot s = (OPt - OAt) \cdot s * s \cdot s = (Pt - At) \cdot s * s \cdot s$$

在连续碰撞检测中,有任意一点 $V_i = V_0 * (1 - t) + V_1 * t$,因此有:

$$atpt \cdot s = ((P0 * (1 - t) + P1 * t) - (A0 * (1 - t) + A1 * t)) \cdot s * s \cdot s = ((P0 - A0) * (1 - t) + (P1 - A1) * t) \cdot s * s \cdot s = (P0 - A0) \cdot s * s \cdot s * (1 - t) + (P1 - A1) \cdot s * s \cdot s * t = (P0 - A0) \cdot s * (1 - t) + (P1 - A1) \cdot s * t$$

其中, $P0, A0$ 表示点 P, A 在 $t=0$ 时刻的坐标,点 $P1, A1$ 表示点 P, A 在 $t=1$ 时刻的坐标, $t \in [0, 1]$ 。因此,在 VF - pair 中只要满足式(1)中 $X1, X2, X3, X4, X5, X6$ 同号,那么此时间间隔内不发生碰撞。

$$\begin{cases} X1 = (P0 - A0) \cdot s \\ X2 = (P1 - A1) \cdot s \\ X3 = (P0 - B0) \cdot s \\ X4 = (P1 - B1) \cdot s \\ X5 = (P0 - C0) \cdot s \\ X6 = (P1 - C1) \cdot s \end{cases} \quad (1)$$

其中, $P0, A0, B0, C0$ 表示点 P, A, B, C 在 $t=0$ 时刻的坐标,点 $P1, A1, B1, C1$ 表示点 P, A, B, C 在 $t=1$ 时刻的坐标。

在 EE - pair 中只要满足式(2)中 $X1, X2, X3, X4, X5, X6, X7, X8$ 同号,那么此时间间隔内不发生碰撞。

$$\begin{cases} X1 = (A0 - C0) \cdot s \\ X2 = (A0 - D0) \cdot s \\ X3 = (B0 - C0) \cdot s \\ X4 = (B0 - D0) \cdot s \\ X5 = (A1 - C1) \cdot s \\ X6 = (A1 - D1) \cdot s \\ X7 = (B1 - C1) \cdot s \\ X8 = (B1 - D1) \cdot s \end{cases} \quad (2)$$

其中, $A0, B0, C0, D0$ 表示点 A, B, C, D 在 $t=0$ 时刻的坐标,点 $A1, B1, C1, D1$ 表示点 A, B, C, D 在 $t=1$ 时刻的坐标。

在实验中发现,采用方向向量为 $(1, 0, 0), (0, 1, 0), (0, 0, 1)$ 3条直线投影和仅使用 $(1, 0, 0)$ 的效果相同,因此对坐标轴上的投影仅用 X 轴。除此之外,还用了方向向量分别为 $(1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)$ 的4条直线投影,比仅使用坐标轴投影效果更好。考虑到投影用的直线越

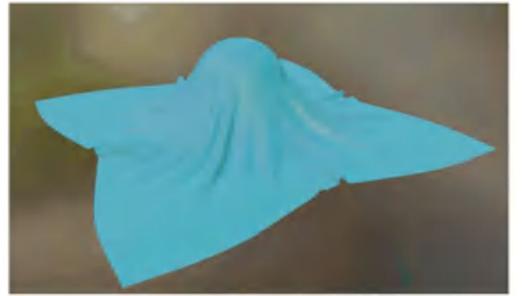
多,计算所花费时间越多,但剔除效果提升不多,故只用这5条直线作为投影用直线,即式(1)和式(2)中 s 的取值有 $(\sqrt{2}/2, \sqrt{2}/2, 0), (\sqrt{2}/2, 0, \sqrt{2}/2), (0, \sqrt{2}/2, \sqrt{2}/2), (\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3), (1, 0, 0)$ 。

3 结果与讨论

本文实验环境为 Intel Core i7-8565U CPU@1.80 GHz 2.00 GHz,编译环境为 VS2017,代码编写采用 C++,测试数据为 funnel 和 cloth_ball 数据模型,数据来自北卡罗来纳大学教堂山分校(University of North Carolina at Chapel Hill, UNC)。funnel:9 450 个点,18 484 个三角面;cloth_ball:46 598 个点,92 230 个三角面,渲染图如图 5 所示。



(a) funnel 渲染图



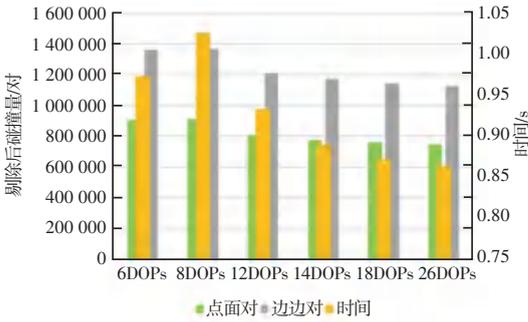
(b) cloth_ball 渲染图

图 5 数据集渲染图

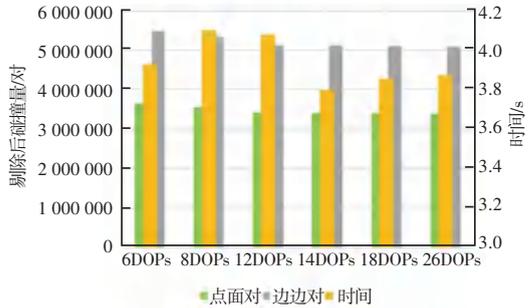
Fig. 5 Rendering image of dataset

3.1 确定 k 值

对比不同包围盒的优劣,本文最终选择 kDOPs 包围盒。为了确定适合的 k 值,本文对 k 为 6、8、14、18、26 分别进行测试,对比不同 k 值在 funnel 和 cloth_ball 数据上一帧的剔除数量与时间,结果如图 6 所示。在 funnel 上, k 为 26 时无论是边边对还是点对面的剔除率都是最高的,且花费时间最短;在 cloth_ball 上, k 为 26 时无论是点面对还是边边对的剔除率都是最高的,时间稍稍长于 k 为 14 和 18 时。综合考虑花费时间与剔除数量后,本文采用 26-DOPs。



(a) funnel 上 k 值测试结果



(b) cloth_ball 上 k 值测试结果

图 6 k 值测试结果

Fig. 6 Results of the k -value test

3.2 滤波器实验

空间线性投影滤波器 (SLPF) 对基本测试中不满足内部条件的碰撞对进行快速剔除, 为提高剔除率, 又增加一个非穿透性滤波器 (DNF) 来进行非共面剔除。分别在 cloth_ball 和 funnel 上进行实验, 经过 26-DOPs 剔除后, 对比仅使用 SLPF、仅使用 DNF 以及使用 SLPF+DNF 进行剔除的效果, 结果见表 1 和表 2, 其中时间为每帧时间。可以看出, 仅使用 SLPF 比仅使用 DNF 的剔除数量更多, 使用 SLPF+DNF 的剔除数量最多。由于 SLPF+DNF 路径是在 SLPF 为真的情况下再进入 DNF 判定, 增加了计算, 所以时间花费比只经过一层滤波器稍多。而对于 SLPF 或者 DNF, 不同的碰撞位置需要计算的不等式数量不同, 而具体哪个滤波器需要计算的不等式数量更多无从判定, 因此有时 SLPF 花费的时间多, 有时 DNF 花费的时间多。

表 1 cloth_ball 上的测试结果

Tab. 1 Test results on cloth_ball dataset

	SLPF	DNF	SLPF+DNF
VF 剔除率/%	58.18	38.46	58.57
VF 时间/s	7.672 4	7.777 36	7.821 36
EE 剔除率/%	45.79	31.23	47.21
EE 时间/s	11.628 61	11.360 65	11.734 03

注: 加粗字体为每行最优值

表 2 funnel 上的测试结果

Tab. 2 Test results on funnel dataset

	SLPF	DNF	SLPF+DNF
VF 剔除率/%	58.88	56.46	63.17
VF 时间/s	1.674 93	1.775 48	1.749 56
EE 剔除率/%	48.01	47.81	53.18
EE 时间/s	2.535 52	2.652 59	2.578 57

注: 加粗字体为每行最优值

4 结束语

本文对连续碰撞检测算法进行研究, 采用两级碰撞检测算法作为框架。高层剔除阶段在分析各种包围盒优劣后, 决定选择 k -DOPs 作为包围盒层次结构的包围盒, 经过实验确定 k 值为 26。由于一对碰撞对需要满足共面条件和内部条件才能确定是发生碰撞, 而已经存在的 DNF 只能剔除不满足共面条件的碰撞对, 于是在低层剔除阶段, 提出一个可以剔除不满足内部条件的碰撞对的空间线性投影滤波器 (SLPF), 结合使用非穿透性滤波器 (DNF), 可以显著提高剔除效率。最后, 在两个数据集上进行实验, 验证了这一理论。实验结果表明, 仅使用 SLPF 比仅使用 DNF 的剔除数量更多, 使用 SLPF+DNF 的剔除数量最多。

本文算法在剔除数量上具有一定优势, 但是在时间花费上并不占据绝对优势, 还有改进的空间, 比如: 简化计算步骤。除此之外, 本文算法是基于 CPU 实现的, 后续可以考虑基于 GPU 的并行策略来进一步减少使用时间。

参考文献

- [1] Ericson C. Real-Time Collision Detection [M]. London: CRC Press, 2004, 553-575.
- [2] ZHANG X, REDON S, LEE M, et al. Continuous collision detection for articulated models using Taylor models and temporal culling [J]. ACM Transactions on Graphics, 2007, 26(3): 15-es.
- [3] REDON S, KHEDDAR A, COQUILLART S. Fast Continuous Collision Detection between Rigid Bodies [J]. Computer Graphics Forum, 2002.
- [4] PROVOT X. Collision and self-collision handling in cloth model dedicated to design garments [C]// Computer Animation and Simulation. 1997: 177-189.
- [5] 杜鹏. 柔性物体连续碰撞检测若干关键技术研究 [D]. 浙江: 浙江大学, 2013.
- [6] TANG M, MANOCHA D, TONG R F. Fast continuous collision detection using deforming non-penetration filters [C]// In Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games (I3D '10). Association for Computing Machinery, New York, NY, USA, 2010: 7-13.