

一种基于分离包围盒的快速碰撞检测算法*

王 伟⁺, 李文辉, 张振花

(吉林大学 计算机学院 符号计算与知识工程教育部重点实验室, 吉林 长春 130012)

Efficient Collision Detection Algorithm Based on Separating Bounding Volumes

WANG Yi⁺, LI Wen-Hui, ZHANG Zhen-Hua

(Key Laboratory of Symbol Computation and Knowledge Engineering for the Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China)

+ Corresponding author: E-mail: wangyi@email.jlu.edu.cn

Wang Y, Li WH, Zhang ZH. Efficient collision detection algorithm based on separating bounding volumes. *Journal of Software*, 2008,19(Suppl.):143–150. <http://www.jos.org.cn/1000-9825/19/s143.htm>

Abstract: An efficient collision detection method based on separating bounding volume (SBV) is proposed. The positions and shapes of SBVs are determined by the optimal separating support hyper planes of two objects. SBVs not only can efficiently detect the separation of models, but have a high culling ratio when models are intersecting. In order to compute SBVs efficiently, an approximate method using SVM is also put forward and tested. At last in penetration region, a method combined with GPU and SBVs is designed to handle the proximity queries. Experimental results illustrate that SBVs based collision detection algorithm is applicable to exact collision detection for 3D models even without topologies and achieves more efficient and balanced performances in separating, colliding and especially puncturing cases.

Key words: collision detection; bounding volumes; separating plane; support vector machines

摘 要: 提出了一种基于分离包围盒(SBVs)的快速碰撞检测方法.SBVs 的空间形态和位置由两个模型的最优分离平面所决定,这使得它不仅可以快速检测出分离模型,而且在模型相交的情况下能够有效地缩小精确检测的范围.为了能够快速计算 SBVs,设计并验证了一种基于 SVM 的近似计算 SBVs 方法.最后将 SBV 和图形硬件的计算优势结合起来,以实现复杂模型相交区的穿刺查询.实验结果表明,基于 SBVs 的碰撞检测算法能够高效、平衡地处理无拓扑模型的分离、碰撞,尤其是穿刺等复杂情况.

关键词: 碰撞检测;包围盒;分离平面;支撑向量机

碰撞检测(collision detection,简称 CD)是机器人、动画仿真、虚拟现实等领域不可回避的问题之一,其基本任务是确定两个或多个物体彼此之间是否发生接触或穿透.近年来,国内外学者对此进行了大量的研究^[1].其中,

* Supported by the National Natural Science Foundation of China under Grant No.60573182 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2008AA10Z224 (国家高技术研究发展计划(863)); the Doctoral Fund of the Ministry of Education of China under Grant No.20060183042 (国家教育部博士点基金); the Jilin Province Sci-Tech Development Plan of China under Grant No.20060527 (吉林省科技发展计划)

Received 2008-05-03; Accepted 2008-11-14

在物体空间中基于层次包围体树(hierarchical bounding volume trees,简称 BVH)类算法有效地解决了刚体的碰撞检测.而对于变形物体来说,BVH 不仅需要预处理过程,而且需要频繁的更新来配合模型的变形运动,所以在复杂动态场景中还不能达到令人满意的效果.而基于图像的碰撞检测技术弥补了这一不足^[2].这类算法一般将三维几何物体通过图形硬件投影绘制到图像平面上,降维得到二维的图像空间,然后通过保存在各类缓存中的信息进行查询和分析,检测出物体之间是否发生干涉.算法优势在于能够有效利用图形硬件加速技术来减轻 CPU 的计算负荷,从而达到提高算法效率的目的.而结合基于图像空间和物体空间的两类碰撞检测算法的混合算法能够利用二者的优点增强算法的功能^[2].这类混合算法一般分 3 个步骤来完成:(1) 初步检测,在物体空间内通过某种几何结构确定潜在碰撞关键区;(2) 由深度缓存和模板缓存信息建立关键区的层次深度图(layered depth images,简称 LDIs);(3) 在 LDIs 中进行详细查询.其中第 2 步的计算涉及 GPU 读取操作比其他两步费时.目前已经有大量的相关研究,如压缩 LDI 算法和负载平衡策略等^[3],但也不能忽视关键区紧密性对整个算法效率的影响,因此,人们通常利用某种包围盒结构或 BVH 来提高关键区的紧密性.但现有的包围盒类型都是面向单个对象构造的,在模型对的相对方向上常常并不紧密,而混合 BVH 的初步检测常常不能很好地剔除不相交区域,更不适合大型的变形物体模型^[4],所以,找到一种全面而又稳定的初步检测方法对于基于图像碰撞检测过程至关重要.

由于判断三维空间中的两个模型的碰撞问题可以转化为判断是否存在一个平面能够将其分开的问题^[5],所以本文在研究分离超平面定理的基础上提出了一种新型的碰撞检测模型——分离包围盒(separating bounding volumes,简称 SBVs).SBVs 是一种成对出现的特殊类型的凸包,它继承了凸包紧密性好的优点,同时它的空间形态和位置是由两个模型的最优分离平面所决定.利用 SBVs 可以快速判定物体是否分离并在模型相交时准确定位物体的相交区域.本文将给出 SBVs 的定义、一种基于支撑向量机(support vector machine,简称 SVM)^[6]近似 SBVs 计算方法以及关键区内基于 LDIs 的通用的穿刺查询算法.最后与已有的算法进行了比较,表明本文算法有许多优秀的特性,能够高效、平衡地处理虚拟环境中无拓扑信息物体模型的分离、碰撞,尤其是穿刺等复杂情况.

1 分离包围盒定义

首先我们给出 SBVs 的定义以及本文中涉及的概念.

定义 1. 设 $d \in \mathbb{R}^n$ 为非零向量, $X \in \mathbb{R}^n$ 为一个非空点集,定义 X 在方向 d 上的最大延伸为 $E_d(X) = \text{Max} \{d^T x \mid x \in X\}$.

定义 2. 根据定义 1, \mathbb{R}^n 中满足条件 $d^T x = E_d(X)$ 的点 x 所组成的集合 $P_d(X) = \{x \mid d^T x = E_d(X)\}$ 称为 X 在方向 d 上的支撑超平面.其中, d 称为支撑超平面的法向量,满足条件 $x \in P_d(X) \cap X$ 的点 x 称为支撑点(support point).

定义 3. 根据定义 2,由 x 在方向 d 上的支撑超平面所定义的 \mathbb{R}^n 中的闭半空间 $H_d(X) = \{x \mid d^T x \leq E_d(X)\}$ 称为 X 在方向 d 上的包围半空间.其中, d 称为 $H_d(X)$ 的法向量.

定义 4. 已知非空凸点集 X_1 与 X_2 和一个向量 v ,令 p 是 X_1 相对于 v 的一个支撑点, q 是 X_2 相对于 v 负方向的一个支撑点,称有序对 (q, p) 为 X_1 和 X_2 相对于 v 的支撑点对,称向量 (q, p) 为 X_1 和 X_2 相对于 v 的支撑向量(support vector,简称 SV).

定理 1(分离超平面定理(separating hyperplane theorem)). 在 \mathbb{R}^n 中,如果非空集合 X_1, X_2 是凸集且没有共同的点,则一定存在平面 $H = \{x \mid p^T x = a\}$ 将集合 X_1, X_2 分开,且有 $\{\forall x_1 \in X_1, p^T x_1 \geq a\}, \{\forall x_2 \in X_2, p^T x_2 \leq a\}$.

分离超平面定理说明能够分离三维空间中两个凸点集的平面要么是唯一的,要么为无数个,但在这些平面中只有 1 个能使两凸点集的分类间隔最大,称其为最优分离平面(optimal separating hyperplane,简称 OSH)^[7],其法向量为最优分离方向向量 d^* . d^* 与 $-d^*$ 和两点集相对于它们的支撑点对所确定的平行平面称为最优分离支撑超平面(optimal separating support hyperplane,简称 OSSH).OSSHs 之间的距离为两凸集间的最大间隔距离.我们定义分离包围盒为包含两 OSSHs 的平行平面对应的集合组成.

定义 5. 设 $D = \{d_1, d_2, \dots, d_{2k}\}$ 为 \mathbb{R}^n 中有 $2k$ 个固定不变的非零向量的集合,且对 $\forall d_i \in D$,必有 $\forall d_j \in D$,使得

$d_j = -d_i$. 令 $D^* = \{d^*, -d^*\}$, 其中, d^* 为非空的凸点集 X_1 与 X_2 的最优分离平面的法向量. 令 $D' = D \cup D^*$, 对于 X_1 和 X_2 , 我们定义其包围半空间的集合构成关于 D' 的分离包围盒为

$$SBV_{D'}(X_1) = \cap (H_d(X_1))_{d \in D'} \tag{1}$$

$$SBV_{D'}(X_2) = \cap (H_d(X_2))_{d \in D'} \tag{2}$$

分离包围盒的方向集 D' 中的方向向量为共线且方向相反的向量对. 当 D' 中包含了 \mathfrak{R}^n 中所有的方向向量时, 发展成为分离凸包. 当 D' 中包含 D^* 和其他两个方向对组成两两垂直的分离坐标轴方向时, 称为分离坐标轴向包围盒, 简记为 SBVs (如图 1 所示). 考虑到分离坐标轴向包围盒计算和相交测试比较简单, 以下均采用这种分离包围盒.

SBVs 与其他包围盒区别在于, 它是从两个模型的位置关系出发, 在模型相对方向上最紧密. 例如图 2 中, 在模型分离的情况下, SBVs 一步就能判断出分离, 而 AABB^[8] 和 OBB^[9] 都需要建立 BVH 才能判别出模型是否分离; 而在物体相交情况下, AABB 和 OBB 包含了较多的模型不相交元素, 增加了详细检测的计算量. 而近似 SBVs (在第 2.2 节给出算法) 不但排除了较多不相交特征, 而且它们的最大优势在于计算的正交区是包含了模型的最优分离方向, 这为下一步详细检测提供了一个良好的基础.

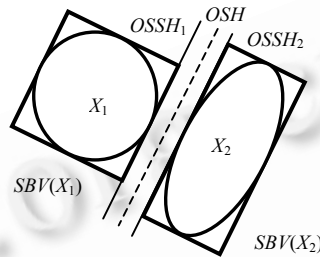


Fig.1 SBVs of two convex sets X_1 and X_2

图1 两个凸点集 X_1 和 X_2 的分离坐标轴向包围盒 SBVs

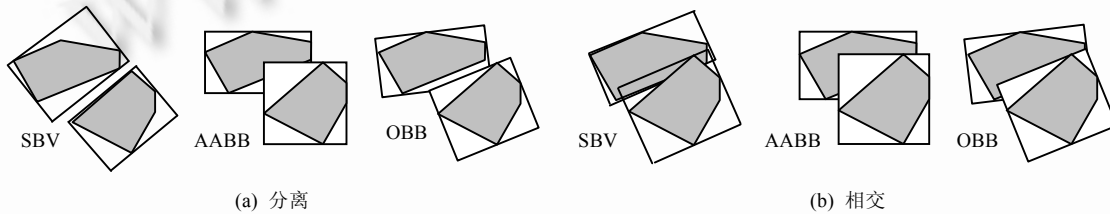


Fig.2 Comparison of the bounding tightness among some bounding volumes and SBVs in separating and intersecting cases

图2 几种常用包围盒与 SBVs 在模型分离和相交情况下紧密性的对比

2 近似 SBVs 计算及关键区生成

由分离包围盒的定义可知, 计算的包围盒的关键在于计算两个分离凸点集的最优分离平面. 由于我们可以把两个模型的特征点集合看作两个特征类, 模型间的最优分离平面即两个特征点集合的最优分离平面, 对于这个问题, 我们采用支撑向量机来进行求解. 而对于非凸或线性不可分模型, 我们也可以通过限制训练样本来实现 SBVs 的快速近似计算.

2.1 SVM

SVM 是 20 世纪 90 年代由 Vapnik 等人提出的一种机器学习方法, 它在解决小样本、非线性及高维模式识别问题中表现出许多特有的优势, 已成为继模式识别和神经网络研究之后机器学习领域新的研究热点^[10,11]. SVM 的目标是根据训练样本构造一个最优分类超平面, 使得样本集可以被该超平面尽可能正确地分开, 并且离超平面最近的向量与超平面之间的距离最大. 对非线性可分数据集, SVM 通过核函数对待分类数据进行非线性特征映射, 将数据映射到某一更高维特征空间中, 从而能够线性可分, 然后在其中构造(广义)最优分类超平面. 简单来说, 对两类分类问题, 设训练样本集为 $(x_i, y_i) (i=1, 2, \dots, n)$, $x_i \in \mathfrak{R}^n$ 为训练样本, $y_i \in \{+1, -1\}$ 是输入样本的类标记, 核函数为 $K(x_i, x_j)$. 分类问题可以表示为二次规划问题:

$$\bar{a} = \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^l \alpha_i,$$

满足:

$$\begin{cases} 0 \leq \alpha_i \leq C, i = 1, \dots, l \\ \sum_{i=1}^l \alpha_i y_i = 0 \end{cases} \quad (3)$$

分类器为 $f(x) = \text{sign}\left(\sum_{x_i \in SV_s} \bar{a}_i y_i K(x_i, x) + \bar{b}\right)$, 其中 a_i 为拉格朗日乘子(Lagrange multiplier), C 为惩罚参数.

对于线性可分 SVM 的核函数为 $K(x_i, y_i) = (x_i \cdot y_i)$. 分类器为 $f(x) = \text{sign}(\bar{w} \cdot x + \bar{b})$, 其中, $\bar{w} = \sum_{i=1}^l \bar{a}_i x_i y_i$, $\bar{b} = \left(\sum_j \bar{a}_j y_j x_j^T x_i - y_i\right)$, $a_i > 0$, 所对应的向量为支撑向量, 两类间的最大间隔距离 $mar = 2\|\bar{w}\|^{-1}$.

与一般分类的问题不同, 物体特征集合代表的两个类是确定的, 利用 SVM 的目的是计算两个特征类的最优分离平面和支撑向量. 如果两个特征集合线性可分, 那么直接利用线性 SVM 训练, 即利用公式(3)进行计算, 求得最优分离平面 $OSH = \{\bar{w} \cdot x + \bar{b}\} = 0$, 支撑平面 $OSSH = OSH \pm \frac{1}{2}mar$. 但如果两模型线性不可分、相交, 甚至相包含, 我们通过简化训练特征使得训练特征为线性可分的, 这样不用调节参数直接使用线性 SVM 训练就能达到令人满意的效果.

2.2 核心算法

令模型顶点集 A, B 表示模型 A 和 B 的初始训练特征集合, 其中 A 为正样本集, B 为负样本集, 并为每个样本加入类别标签 $\{-1, 1\}$. 把每个样本的三维坐标看作分类训练的 3 个特 $Features = [X, Y, Z]$. 为了保证数值计算的稳定性, 常对特征数值进行尺度调整正则化到 $[-1, +1]$ 区间.

考虑到不是所有特征都决定两个模型位置关系, 所以找到那些“有价值”的特征能够加快 SBV 的计算速度. 因此, 我们首先令两个模型的 SBVs 初始方向集为世界坐标系坐标 $D_0 = \{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}$, 然后挑选出由 D' 决定的两模型 SBVs 的支撑点集 SV_a 和 SV_b , 中线性可分的点作为模型的初始训练特征, 这样训练集合既是凸点集又是线性可分的, 从理论上一定存在最优分离平面. 之后由 SVM 训练逐步逼近由模型对最优分离平面决定的分离包围盒, 并返回模型之间的位置关系. 因此, 两模型 SBVs 的计算过程也就是初步判断两个物体位置关系确定相交区域的过程(如图 3 中 $\text{Intersection}(A, B)$), 整个近似计算算法描述如图 4 所示.

在算法中建立 SBVs 的时间, 取决于模型的特征数量, 计算复杂度为 $O(n)$. SVM 的训练时间, 取决于训练样本的数量, 计算复杂度为 $O(n^2)$ ^[11]. 由于我们利用 SBVs 的支撑特征作为样本训练, 在一般情况下, 其远远小于模型的特征数量, 所花费的时间并不多, 所以近似计算 SBVs 的算法可以近似在线性时间内完成.

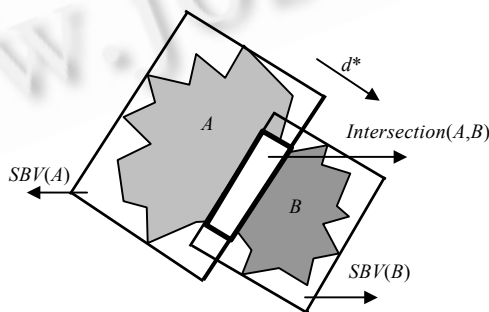


Fig.3 An example for the approximate computation algorithm for SBVs

图3 一个SBVs近似计算例子

```

CSBV(A,B){
  Build(SBV(A),SBV(B),D0);
  if (SVa∈SBV(B)∪(SVb∈SBV(A))=true then
    Intersection(A,B)=SBV(A)∪SBV(B);
    PenetrationSearch(Intersection(A,B),A,B);
  End if
  Else
    OS(A)={a|a∈SVa∩a outside SBV(B)};
    OS(B)={b|b∈SVb∩b outside SBV(A)};
    D'=SVM(OS(A),OS(B))
    Compute(SBV(A),SBV(B),D');
    I(A)={a|a∈A∩a inside SBV(A)∩SBV(B)};
    I(B)={b|b∈B∩b inside SBV(A)∩SBV(B)};
  Repeat
    Compute(SBV(I(A)),SBV(I(B)),D');
    I(A)={a|a∈A∩a inside SBV(A)∩SBV(B)};
    I(B)={b|b∈B∩b inside SBV(A)∩SBV(B)};
  Until (SBV(I(A))∩SBV(I(B))=null or (SBV(I(A))=SBV(I(B))))
    I*(A)={a|a∈A∩a inside SBV(I(A))};
    I*(B)={b|b∈B∩b inside SBV(I(B))};
    If SBV(I(A))=SBV(I(B)) then
      Intersection(A,B)=SBV(I(A));
    End if
    If SBV(I(A))∩SBV(I(B))=null then
      D'=SVM(I*(A),I*(B))
      Compute(SBV(A),SBV(B),D');
      Intersecting=false;
    End if
  End Else
} End CSBV(A,B)

```

Fig.4 Approximate computation algorithm for SBVs

图4 SBV近似计算算法描述

此外,由近似算法 CSBV 得到的近似 SBVs 与由两模型最优分离平面确定的 SBVs 之间存在一些方向上的误差,但对于分离的情况来说,这个误差并不影响判断出模型没有碰撞的结果.对于相交来说,它们都能正确地圈定物体的相交区域,只不过分离方向和进入圈定区域的未碰撞特征不同.这个差异可在每个时间片运动更新过程中逐步减少.由于时空相关性和耦合性,支撑向量在每个时间片位置变化较小,通过把上一个时间片 SBVs 的支撑向量集合作为这个时间片的输入样本,更有利于 SVM 训练,使 SBVs 圈定的区域更加逼近真实的相交区域.

3 详细检测

在关键区确定之后,需要进一步判断两模型是否相交.在此我们可以通过图形硬件对相交区由顶点 $I^*(A)$ 和 $I^*(B)$ 确定的三角面片集 A' 和 B' 的绘制过程来实现.分别选取矩形相交区域 $Intersection(A,B)$ 三个正交的方向为采样方向(Z),并在这些方向上分别进行基于层次深度图像的穿刺查询.

层次深度图像 LDIs 是一帧每个离散像素位置上包含多个深度像素的图像,LDIs 的每个像素包含 5 个分量,3 个分量 R,G,B 表示像素的颜色,分量 A 表示像素透明度,分量 D 表示像素的深度.对于位于同一深度的若干像素,可采用简单的平均融合得到该深度像素的颜色值.对于 LDI 的生成,我们采用文献[2]中的算法利用深度缓

冲和模板缓冲多路并行计算,并利用硬件加速实现深度缓冲的排序,即按视线方向为模型在 LDIs 中每个像素深度队列中的片段按深度值进行成对的排序,得到有规律的 LDIs,之后并行的在每层相交区 $LDI^*=LDI(A)\cap LDI(B)$ 中分别判断每个像素上的深度队列中是否有两模型的片段相交,如果相交就检测到了一个碰撞对,详见文献[2].整个穿刺查询算法流程如图 5 所示,而整个检测过程如图 6 所示.

```

PenetrationQuery(A,B) {
    RenderVolume=Intersection(A,B)
    Update(A',B') to GPU
    LDIs(A)=Render(A')
    LDIs(B)=Render(B')
    Read back the LDIs to the CPU;
    Return Intersection(LDI(A'),LDI(B'))
}
    
```

```

Initial Stage {
    Compute SBV(A)
    Compute SBV(B)
}
Simulation Circle {
    Update(A,SBV(A))
    Update(B,SBV(B))
    CSBV(A,B)
    PenetrationQuery(A,B)
    Return collision information
}
    
```

Fig.5 LDIs based Penetration Query algorithm

Fig.6 Whole collision detection process

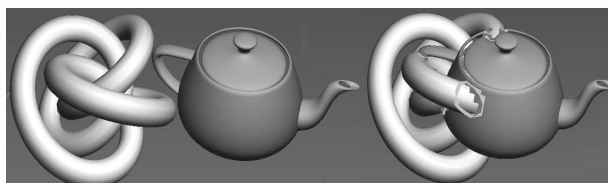
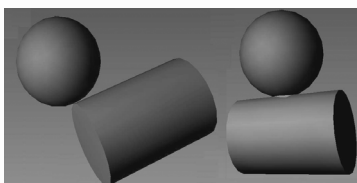
图 5 基于 LDIs 的穿刺查询算法

图 6 整个碰撞检测算法流程

4 实验与性能分析

我们在 PC(CPU 2.2, RAM 1G, NVIDIA Geforce 6800, 显存 256M) 上用 VC++ 实现了本文提出的基于 SBVs 的碰撞检测算法(简称 SBV). SVM 训练由 LIBSVM(2.84) 支持^[11]. 对比实验采用基于 AABB 层次体树碰撞检测算法的 Opcode 包^[12]并加入文献[13]中提出的混合层次体树更新过程(简称 AABB).

首先,我们在静态场景中测试 SBVs 的计算速度和定位碰撞区域的精确度.测试分为两组(如图 7 所示): (a) 圆柱和球体; (b) 绳结与茶壶.测试结果见表 1.



(a) Scene 1: The cylinder is of 3 000 primitives and the ball is of 3 000 primitives

(b) Scene 2: The knot is of 7 000 primitives and the teapot is of 10 000 primitives

(a) 场景 1: 圆柱和圆球的碰撞, 均采样 3 000 个特征

(b) 场景 2: 绳结和茶壶的碰撞, 分别采样 7 000 和 10 000 个特征

Fig.7 Collision detection for rigid models in static scene

图 7 静态场景中刚体的碰撞检测

Table 1 Some parameters of SBVs in separating and intersecting cases

表 1 静态场景中两组模型在分离和相交情况下 SBVs 的有关参数

Scene	Separating		Intersecting		Culling ratio (%)
	TS	T_{svm} (ms)	TS	T_{svm} (ms)	
(a)	290+5	8	288+3	8	95
(b)	153+65	7	119+38	7	86

表 1 中 TS 是 SVM 的训练样本数; Culling ratio 是在 $Intersection(A,B)$ 内的碰撞对与总的特征对数的比; T 为 SVM 训练时间. 由于 SVM 只训练支撑特征, 输入样本比较少, 所以整个计算 SBVs 的时间花费不多. 而且近似算

法求得的 SBVs 的对刚体碰撞的定位碰撞能力也较强,无论凸凹,在相交区域内的碰撞率都达到 85%以上,为穿刺查询提供了一个很好的基础。

对于一些特殊情况,例如支撑点过多导致训练时间过长,我们采取采样部分特征的办法减少训练集.以(b)组模型线性可分的情况为例,假设模型上所有特征都为支撑点,分别采样几组支撑点的作为输入样本.在表 2 中,SV number 是支撑向量的数目,Training time 为 SVM 训练时间,Differ 为由样本训练出的分离方向 d 与模型最优分离方向 d^* 角度之间的差的绝对值.从表中结果可以看出,几组样本经过线性 SVM 训练都能达到 100% 的分类正确性,而且训练时间也可以接受.此外,由样本训练出的分离方向与最有分离方向之间的差仅取决于支撑向量的数目,所以尽可能地采样支撑向量有助于提高训练速度和精确性。

Table 2 Training result for sampling SV of the knot and the teapot in Fig.7(b)

表2 对图7(b)中绳结和茶壶采样部分支撑点训练结果

Samples	SV number	Training time (ms)	Differ (°)
100+100	37	6	10
250+250	80	11	4
500+500	93	23	2
2 500+2 500	111	105	2
7 000+10 000	131	387	0

场景 3 模拟的是在 100 个时间步(每个时间步不设上限)内手抓球的动态效果,如图 8 所示.从图 9 中可以看出,即使采用了混合更新层次体树等优化更新的方法,基于 AABB 层次体树算法也比基于 SBV 的混合算法要慢.SBV 计算简单所需存储空间较少,能够快速排除不相交区域,在相交测试阶段,又可以利用 GPU 的并行性进行计算,从而提高了检测速度。

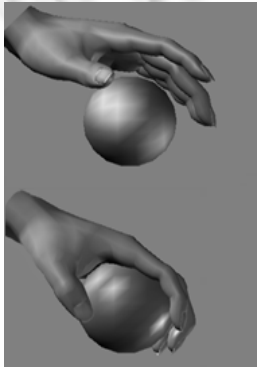


Fig.8 Scene 3: Collision detection for deformable models

图 8 场景 3:变形物体的碰撞检测

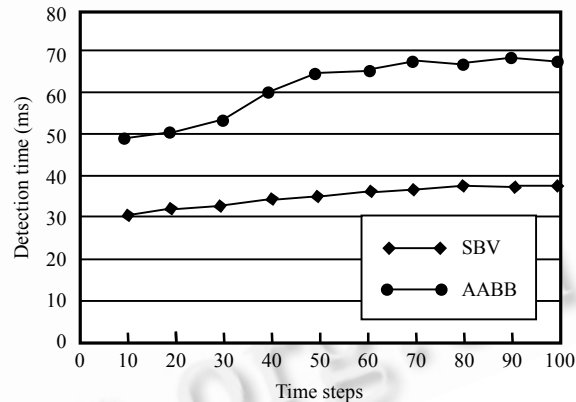


Fig.9 Compare detection time in every time steps between AABB and SBV in Scene 3

图 9 对比场景 3 中每个时间片中 AABB 和 SBV 的检测时间

5 结束语

本文提出了一种基于分离包围盒的快速碰撞检测算法.SBV 最为突出的特点是确定的相交区域内的特征对是具有方向性的.实验表明,在初步检测阶段,SBVs 可以尽早地排除不必要的相交测试,从而提高检测速度.在相交测试阶段,又可以利用 GPU 的并行性进行计算,从而提高计算速度.此外,利用空间的相关性,算法可提高连续运动物体的碰撞检测效率。

References:

- [1] Lin MC, Gottschalk S. Collision detection between geometric models: A survey. In: Proc. of the IMA Conf. on Mathematics of Surfaces. 1998. 37–56.
- [2] Heidelberger B, Teschner M, Gross M. Detection of collisions and self-collisions using image-space techniques. Journal of WSCG, 2004,12(3):145–152.
- [3] Duan JG, Li J. Compression of the layered depth image. IEEE Trans. on Image Processing, 2003,12(3):235–372.
- [4] Govindaraju NK, Knott D, Jain N, Kabul I, Tamstorf R, Gayle R, Lin MC, Manocha D. Interactive collision detection between deformable models using chromatic decomposition. ACM Trans. on Graphics, 2005,24(3):991–999.
- [5] Li XQ, Meng XX, Wang JY, Wang WP, Chung K, Yiu SM. Detecting collision of polytopes using a heuristic search for separating vectors. Chinese Journal of Computers, 2003,7(7):837–847 (in Chinese with English abstract).
- [6] Burges CJC. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 1998,2(2): 121–167.
- [7] Wei YM, Wang Y, Wu QY, Shi JY. Research on fixed direction hull bounding volume in collision detection. Journal of Software, 2001,12(7):1056–1063 (in Chinese with English abstract).
- [8] Bergen VD. Efficient collision detection of complex deformable models using AABB trees. Journal of Graphics Tools, 1997,2(4): 1–13.
- [9] Gottschalk S, Lin MC, Manocha D. OBB tree: A hierarchical structure for rapid interference detection. In: SIGGRAPH'96 Conf. Proc., Annual Conf. Series, ACM SIGGRAPH. New Orleans, 1996. 171–180.
- [10] Li JM, Zhang BL, Fu Z. Training algorithms for support vector machines. Journal of Tsinghua University (Science and Technology Edition), 2003,43(1):120–124 (in Chinese with English abstract).
- [11] Fan RE, Chen PH, Lin CJ. Working set selection using the second order information for training SVM. Journal of Machine Learning Research, 2005,6:1889–1918.
- [12] Terdiman P. Optimized Collision Detection Library. 2003. <http://www.codercorner.com/Opcode.htm>
- [13] Larsson T, Akenine-Möller T. Collision detection for continuously deforming bodies. In: Proc. of the Eurographics. 2001. 325–333.

附中文参考文献:

- [5] 李学庆,孟祥旭,汪嘉业,王文平,Chun GK,Yiu SM.基于启发式搜索分离向量的凸多面体碰撞检测.计算机学报,2003,7(1): 837–847.
- [7] 魏英梅,王涌,吴泉源,石教英.碰撞检测中的固定方向凸包围盒的研究.软件学报,2001,12(7):1056–1063.
- [10] 李建民,张敏林,福宗.支持向量机的训练算法.清华大学学报(自然科学版),2003,43(1):120–2124.



王玮(1980—),女,辽宁沈阳人,博士生,主要研究领域为计算机图像学,虚拟现实.



张振花(1972—),女,博士生,讲师,主要研究领域为计算机仿真,图像检索,多媒体技术.



李文辉(1961—),男,博士,教授,博士生导师,主要研究领域为计算机图形学,虚拟现实,计算机辅助设计,图像处理.