

doi: 10.3969/j.issn.1006-1576.2011.12.023

机器鱼 2D 仿真平台碰撞检测算法

任静, 谢广明

(北京大学工学院, 北京 100871)

摘要: 为了实现机器鱼 2D 仿真平台各个对象间完整的碰撞检测, 提出一种基于离散碰撞检测的算法。该算法以机器鱼 2D 仿真平台为基础, 结合包围盒树, 运用计算机图形学对平台中各个仿真对象进行精确建模; 并利用图形学中的分离轴理论、Voronoi Region 等概念, 提出了适合各对象的不同碰撞检测算法, 快速准确地输出检测结果。同时采用运动插值技术, 减小步长并键入中间运动模态, 实现了模拟连续碰撞检测, 进一步提高了碰撞检测准确度。实验结果证明: 该方法实现了模拟连续碰撞检测, 具有高效、精准的特点。

关键词: 仿真平台; 机器鱼; 碰撞检测; 包围盒; 分离轴

中图分类号: TP306 **文献标志码:** A

Collision Detection on Robot Fish 2D Simulation Platform

Ren Jing, Xie Guangming

(College of Engineering, Peking University, Beijing 100871, China)

Abstract: In order to achieve the all objects collision detection of robot fish 2D simulation platform completed, we propose a collision detection algorithm based on discrete. The algorithm taking 2D simulation platform as base, combined with bounding box tree, use computer graphics put up accurately models to every simulation object; and put forwards different collision detection algorithms adapt to objects use separation axis theory, Voronoi Region concepts in graphics, the output test results quickly and accurately. At the same time using motion interpolation, reducing the step size and type intermediate movement mode to achieve simulation continuous collision detection, further improved collision detection accuracy. Experimental results show that the method can achieved simulating continuous collision detection, has high efficiency, precise features.

Keywords: simulation platform; robotic fish; collision detection; bounding-volume; separating axis

0 引言

近年来, 随着计算机技术和机器人技术的快速发展, 机器人仿真系统应运而生。仿真系统中最基础和核心的内容是对象之间的碰撞处理, 包括碰撞检测和碰撞响应, 前者和计算机技术密切相关, 而后者则主要涉及力学和动力学相关领域。

目前对二维场景中的碰撞检测研究非常广泛, 从时间域的角度划分, 可分为连续碰撞检测、离散碰撞检测和静态碰撞检测等; 从空间域的角度划分, 可分为计算机图形学法、有限元分割法等^[1]。机器鱼 2D 仿真平台中对象趋于多元化, 且对于机器鱼这种多关节驱动的运动模式来说, 只依赖某一种碰撞检测方法无法满足场景模拟需求。因此, 笔者结合计算机图形学等理论, 提出机器鱼 2D 仿真平台各个对象间完整的碰撞检测算法。

1 机器鱼 2D 仿真平台简介

机器鱼 2D 仿真平台 URWPGSim2D

(underwater robot water polo game simulation 2D edition) 是基于 MRDS 的 CCR 和 DSS, 采用 C/S 模式建立起来的, 模拟并仿真实体机器鱼及水环境, 包括环境信息、驱动形式、运动模态、数据流传输形式和外形、游动效果等。

平台仿真周期为 100 ms, 在每个仿真周期中, 客户端加载用户策略, 通过服务器端的公共接口, 传入数据信息, 并进行动力学分析、碰撞处理和实时绘图等。碰撞处理流程如图 1。

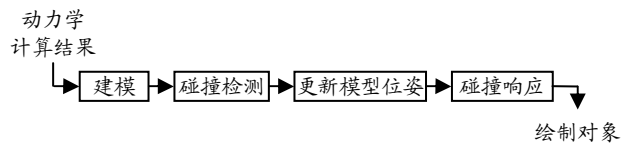


图 1 碰撞处理流程

2 水环境下的对象建模

2.1 环境对象

如图 2, 在机器鱼 2D 仿真平台中, 对象包括仿

收稿日期: 2011-08-15; 修回日期: 2011-09-23

基金项目: 国家自然科学基金项目(10972003); 机器人技术与系统国家重点实验室开放基金项目(SKLRs-2009-MS-09)

作者简介: 任静(1987—), 女, 山西人, 硕士研究生, 从事智能控制研究。

真机器鱼、仿真水球、障碍物、池壁区域等。按照环境对象的状态属性，分为动态(dynamics)和静态(statics)2类。静态类包括障碍物、池壁边界；动态类包括仿真机器鱼、仿真水球。按照环境对象的形状，可分为直线(line)、圆形(circle)、多边形(polygon)等。结合以上的2种分类，定义各个环境对象，方形障碍物(reo)、圆形障碍物(roo)、仿真水球(ball)、仿真机器鱼(fish)，都属于对象集合(obj)， $Obj=\{obj_1, obj_2, obj_3, \dots, obj_n\}$ ($n \in \{1,2,3, \dots, +\infty\}$)；另外，池壁边界 Border= {topline, bottomline, leftline, rightline, goalblock₁, goalblock₂, goalblock₃, goalblock₄}。每个环境对象都分属于静态或动态、直线或圆形或多边形，不同对象的碰撞检测方法会有所不同。

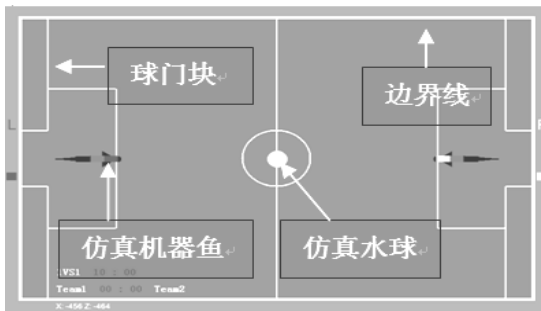


图 2 URWPGSim2D 平台环境场景

现有的 2D 建模方式主要有图形学法和有限元分割法等。针对水中机器鱼 2D 仿真平台实时性要求较高的特点，综合考虑平台的运算速度和效率，笔者主要采用图形学中的包围盒进行建模。常用的包围盒类型如图 3，主要有 Sphere(包围球)、AABB、OBB、K-DOP 等。平台中圆形对象物体的形状以及位置均固定，所以采用最基本的包围球建模。为了提高建模效率，非圆形对象选择包围球或 AABB 模型，为了减少计算机的运算处理量和运算复杂度，同时保证碰撞检测的精度和效率，采用外层模糊，内层精确的两层建模模式。各对象模型(仿真机器鱼除外)属性详见表 1。

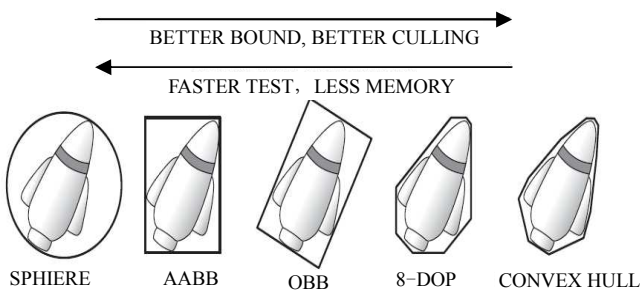


图 3 包围盒的类型^[2]

表 1 各对象模型属性(不包括仿真机器鱼)

| 对象名称 | 状态属性 | 外层模型 | 内层模型 |
|-------|------|------|------|
| 仿真水球 | 动态 | 包围球 | — |
| 圆形障碍物 | 静态 | 包围球 | — |
| 方形障碍物 | 静态 | 包围球 | AABB |
| 球门块 | 静态 | 包围球 | AABB |

2.2 鱼体建模

早在 1996 年 Gottschalk 就提出了 OBB 树在快速碰撞检测中的应用，在此基础上，纽约州立大学的 Klosowski 等人提出了基于 K-DOPs 的层次包围盒，提高了包围紧密度^[3]。在包围盒的所有模型中，最省时而且消耗资源最少的模型是包围球，但其包围紧密性较差。

机器鱼是由刚性躯体、2 个胸鳍、三关节尾部和尾鳍组成。仿真机器鱼由 3 个关节拟合鱼体波运动推进，属于多关节连接连续运动的刚体，机器鱼实体图和仿真图如图 4。如果采用单一的包围盒建模方式，有可能出现凹多边形的情形，其碰撞检测非常复杂。笔者根据仿真机器鱼多关节的特点，结合包围球的高效快速优点和 k-d 树良好的包围紧密性，采用 BV 树(bounding volume tree)对多关节的仿真机器鱼建模。既满足了其多关节的灵活性，又优化了仿真平台的运算速度。

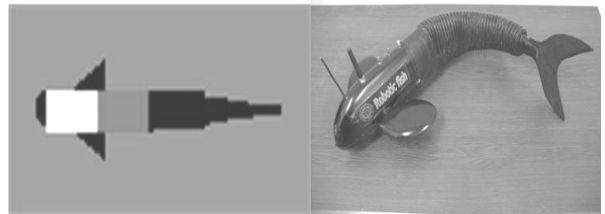


图 4 实体机器鱼和仿真机器鱼对比图

如图 5 和图 6 所示，建立深度为 3 的无序 BV 树。树的叶结点采用 K-DOP 建模，根结点以及子结点均采用包围球模型。在 BV 树的每个结点处计算是否发生碰撞，如果发生碰撞，返回 true，并继续更深层的检测。遍历 BV 树的所有结点，直到返回值为 false 或所有结点检测完成后结束。

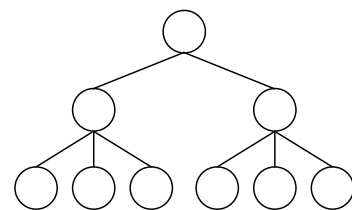


图 5 仿真机器鱼建模树形图

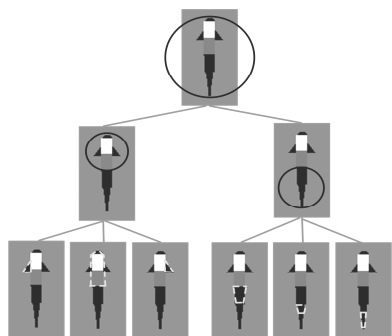


图 6 仿真机器鱼建模示意图

3 拟连续碰撞的设计

按照时间流的角度来划分碰撞检测, 主要分为连续碰撞检测 (CCD) 和离散碰撞检测。对于大量快速运动的小型对象而言, CCD 方法以其准确度高、检测周期短等优点, 可以很好地解决碰撞检测中的难点, 但是这种方法运算耗时过长、占用过多资源, 不能很好地适应实时性要求较高的系统, 这也是 CCD 方法没有得到广泛运用的原因之一^[4]。

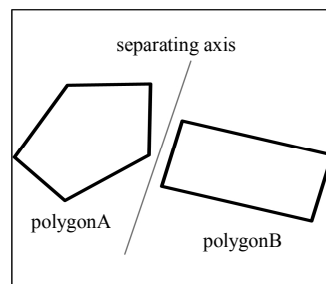
笔者采用离散碰撞检测方法, 并利用运动插值技术, 在每个仿真周期的间隔内插入中间运动模式, 达到模拟连续碰撞检测的效果。机器鱼的状态包括鱼体中心的位置坐标和身体朝向, 由于每个周期机器鱼的尾部都会根据不同的指令呈现不同的状态, 故很难预测鱼的姿态, 所以这里插值均是以当前周期相同模态为基准, 进行状态回退, 实现同模态下的模拟连续碰撞检测。

4 碰撞检测算法

4.1 检测算法

在有限的水池环境里, 首先需要考虑如何检测一个对象是否在此区域中。笔者提出一种简单高效的算法, 即在内存中开辟一个列表空间, 用来存储碰撞模型中的极值坐标, 检测所有极值坐标是否均在水池边界中。

在被检测对象都位于水池边界范围内时, 进一步考虑如何检测多边形之间是否发生碰撞。仿真平台中对象建模主要涉及到 K-DOP, AABB 和包围球等, 都属于计算机图形学中的基本模型。分离轴是将碰撞问题转换为图形学的关键。如图 7, 所谓分离轴理论, 即在一个二维平面中, 如果 2 个凸多边形不相交, 则一定至少存在一条直线使得这 2 个凸多边形分别位于这条直线的两侧, 则称这条直线为分离轴。由上述理论可知, 寻找被检测物体间是否存在分离轴是碰撞检测的关键。



polygon 为多边形; separating axis 为分离轴。

图 7 分离轴理论

结合 voronoi region 理论和投影法等几何算法, 给出包围盒模型碰撞检测算法如下:

1) 检测包围球是否是多边形的 voronoi region 内^[5], 如果在此区域中, 则跳出算法循环, 并检测 2 个对象的中心距离;

2) 设多边形的边集合 $N = \{n_1, n_2, n_3, \dots, n_m\}$ 寻找各边元素的正交向量 $n_i \cdot n_i = 0$, 示意图可参照图 8;

3) 把多边形投影到各个正交向量上, 得出各个多边形的投影线段;

4) 如果所有多边形投影线段都相交, 则 2 个多边形相交; 反之, 如果存在一种情况, 使得 2 个多边形投影线段没有交集, 则存在一条分离轴, 根据分离轴理论, 2 个多边形在二维平面上不相交, 即 2 个对象没有发生碰撞。

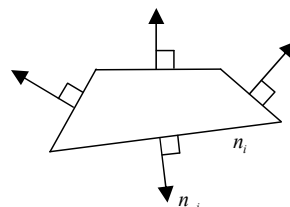


图 8 多边形各边正交向量示意图

4.2 输出参数

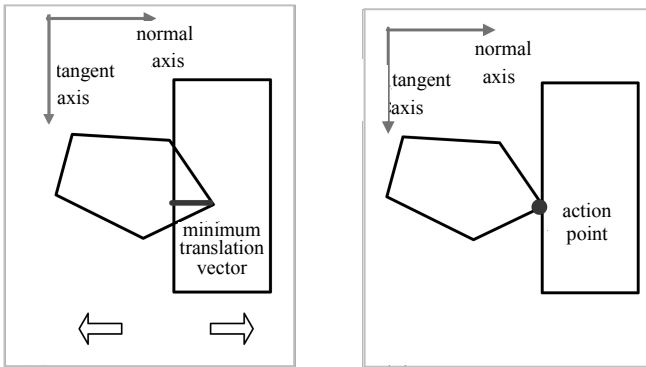
碰撞检测的输出参数存储在一个结构体中:

```
public struct CollisionDetectionResult
{
    public bool Intersect;
    public xna.Vector3
        MinimumTranslationVector;
    public xna.Vector3 NormalAxis;
    public xna.Vector3 ActionPoint;
}
```

由于在上述碰撞检测算法中, 是否发生碰撞的检测结果滞后于真实情况, 输出参数中的最小相交向量 (minimum translation vector) 显示出碰撞发生的深度, 如图 9(a); 法线向量 (normal axis) 在碰撞响应的处理过程中非常重要, 它是最小相交向量发

生的情况下的正交向量，可由法线向量和与之垂直的切线向量确定对象的本地坐标系；在法线向量上，沿着两被检测对象运动的相反方向移动最小相交向量长度，可得碰撞作用点 (action point)，如图 9(b)。

上述碰撞检测算法时间复杂度为 $O(n)$ ，效率较高，占用资源相对较少。当作用对象较多时，如 4 条机器鱼、9 个水球、4 个障碍物以及各个边界，同时进行碰撞检测时，检测所用时间非常少，与 2 个对象碰撞检测所占用的时间几乎相同，并不影响平台的整体运行。



tangent axis 为切线向量；normal axis 为法线向量；minimum translation vector 为最小相交向量；action point 为碰撞作用点。

(a) 碰撞检测前 (b) 碰撞检测后

图 9 碰撞检测前后被检测对象状态

5 结束语

通过上述的建模方式以及各个对象之间的碰撞检测方法，可以很好地适应平台的实时性要求。并且在多物体碰撞(超过 20 个检测对象)的情况下，该算法也可以较为快速精准地完成碰撞检测。

参考文献:

[1] 范昭炜. 实时碰撞检测技术研究[D]. 浙江: 浙江大学, 2003.

[2] Erison C. Real-Time Collision Detection[M]. USA: Morgan Kaufmann, 2005[3] S.Gottschalk, M.C.Lin, D.Manocha. OBBTree: A Hierarchical Structure for Rapid Interference Detection. Proceedings of ACM international Conference on Computer Graphics and Interactive Techniques, 1996: 171-180.

[3] Klosowski J T, Held M, Mitchell J S B, et al. Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs[J]. IEEE Transactions on Visualization and Computer Graphics, 1998, H4(1): 21-36.

[4] 黄通浪, 唐敏, 董金祥. H-一种快速精确的连续碰撞检测算法 H[J]. 浙江大学学报: 工学版, 2006, 40(6): 1051-1055.

[5] Lin M C, Canny J F. A Fast Algorithm for Incremental Distance Calculation[C]. Proceedings of IEEE International Conference on Robotics and Automation, 1991: 1008-1014.

科技教育出版社, 2004: 3-129.

[6] Dorigo, Maniezzo, Colomi. AntSystem: optimization by a colony of cooperating agents[J]. IEEE Transaction on SMC, 1996, 26(1): 8-14.

[7] 辜吟吟. 基于蚁群算法的机器人路径规划方法的研究[D]. 沈阳: 沈阳理工大学, 2007.

[8] Auzinger D, Djumlija G. Application of advanced technology packages for improved strip profile and flatness in hot-strip-mills[J]. Iron and steel, 2004, 39(11): 132-133.

[9] 吴丽娟. 基于机器人足球比赛的进攻策略的实现[J]. 微计算机应用, 2002, 23(6): 351-353.

[10] 中国水中机器人大赛官方网站. URWPGSim2D 开发人员手册[Z]. 北京: 北京大学智能控制实验室, 2010. <http://robot.pku.edu.cn>.

(上接第 86 页)

参考文献:

[1] 岳富占, 崔平远. 基于改进蚁群算法的星球探测机器人路径规划技术 [J]. 控制与决策, 2006, 21(12): 1437-1440.

[2] 王鹏飞, 田冲. 用于多机器人路径规划的一种改进蚁群算法[J]. 电脑知识与技术, 2008, 4(2): 54-55.

[3] 杜治君, 陈姗姗, 李波波. 机器鱼速度控制算法研究[J]. 学术探讨, 知网空间. 2009, (8): <http://www.cnki.com.cn/Article/CJFDTotal-SYWX200908257.htm>.

[4] 柴仲明, 禹梅, 刘舒. 基于遗传算法的机器鱼路径规划 [J]. 兵工自动化, 2010, 29(11): 92-96.

[5] 吴启迪, 汪镭. 智能蚁群算法及应用[M]. 上海: 上海