

移动机器人 3D 仿真软件的设计与实现

刘润丹¹, 王华¹, 付根平²

(1. 阳江职业技术学院 机电系, 广东 阳江 529566;

2. 广东工业大学 自动化学院, 广东 广州 510090)

摘要: 在机器人技术研究中, 为了提高机器人控制算法的开发效率, 提出移动机器人三维仿真软件的设计方案并加以实现。该软件采用 ODE 物理引擎生成动力学世界和实现碰撞检测, 提高了仿真速度和精确度, 同时采用 OpenGL 绘制三维图形, 改善了图形显示效果。仿真实例证明, 该软件具有一定的实用价值。

关键词: 移动机器人; 三维仿真; ODE; OpenGL; 控制算法

中图分类号: TP242

文献标识码: A

文章编号: 1674-7720(2011)04-0004-04

Development of 3D simulation software for mobile robot

Liu Rundan¹, Wang Hua¹, Fu Genping²

(1. Department of Mechanical and Electrical, Yangjiang Vocational and Technology College, Yangjiang 529566, China;

2. College of Automation, Guangdong University of Technology, Guangzhou 510090, China)

Abstract: In robotics research, in order to improve the development efficiency of the robot control algorithm, three-dimension simulation software is designed and implemented for mobile robot. The software uses open dynamics engine to generate dynamic world and realize collision detection, it is helpful to improve the simulation speed and accuracy for using ODE, while three-dimensional graphics are drawn on the screen with OpenGL for the purpose of improving graphic display. The simulation example shows that the software has a certain practical value.

Key words: mobile robot; 3D simulation; ODE; OpenGL; control algorithm

在研究移动机器人控制算法和多机器人协作控制算法时, 常采用仿真软件来提高算法的开发效率并减少实验成本。仿真软件可直接利用物理引擎对刚体对象的运动进行仿真, 物理引擎为刚体赋予真实的物理属性, 以此对物体的运动、旋转和碰撞进行仿真计算。目前主流物理引擎有 Havok (Havok Game Dynamics)、PhysX (Physics Engine Middleware)、Bullet (Bullet Physics Engine) 和 ODE (Open Dynamics Engine) 等^[1], Havok 是业界功能最全面的物理仿真解决方案, 主要应用于实时碰撞计算、动力学约束求解和车辆综合解决方案等领域; PhysX 的特点是适合计算量比较大的物理模拟计算; Bullet 的特点是具有最先进的碰撞检测技术, 主要用于柔体动力学和刚体动力学的仿真; ODE 的特点是具有高级的关节类型和碰撞检测整合系统, 它适合于对机电一体化设备和移动机器人的模拟, 并且易于与 C/C++ API 配合使用, 具有快速性、强健性和可移植性等特点^[2]。

在采用物理引擎生成动力学世界之后, 需利用图形绘制程序将场景绘制到显示终端^[3]。常用的图形绘制函数库有 OpenGL (Open Graphics Library)、DirectX (Direct Extension) 和 Irrlicht (Irrlicht Engine) 等, OpenGL 主要应用于专业高端绘图领域^[4], 具有使用简便、效率高和可移植等特点; DirectX 是微软开发的三维绘图软件包, 性能上已经超过 OpenGL, 但其接口采用 COM 方式; Irrlicht 为高性能实时 3D 引擎, 它可以提供在商业级 3D 引擎上才具有的艺术特性。

综上所述, 在不影响软件性能的前提下, 为了缩短仿真软件的开发周期, 本文采用 ODE 物理引擎和 OpenGL 来开发移动机器人三维仿真软件。

1 软件特性

(1) 采用基于面向对象技术实现, 软件操作简单, 易于维护和功能扩展;

(2) 可以导入 x 格式和 3ds 格式的三维模型文件;

《微型机与应用》2011 年 第 30 卷 第 4 期

(3) 允许物体同时实现多个移动操作, 在每个运动方向都有加速度、减速度和最大速度等运动属性; 实现碰撞检测、移动机器人和虚拟场景的图形化显示;

(4) 支持实时调试功能; 3D 动画和仿真计算结果同步且真实对应; 绘制仿真环境的二维地图和物体运动轨迹。

(5) 提供与外部软件连接的接口, 即可以通过 ODBC 与外部数据库相连, 或通过 Socket 接口与外部设备相连, 实现进程之间的通信;

(6) 软件接口的多样性和扩展性, 即可通过游戏手柄、键盘来输入控制信息、模型参数和仿真参数等; 仿真数据的保存输出。

2 仿真软件的关键技术

2.1 移动机器人的速度分解公式

对于地面移动机器人平台, 可根据车体的运动方式和实验数据求取其速度分解公式。本文采用由三个全向轮驱动的移动机器人^[5], 规定机器人坐标系是以机器人的正前方为 x 轴正方向, 逆时针旋转 90° 为 y 轴正方向, 并以三个轮系的机械中心为原点, 机器人坐标系及速度分解如图 1 所示。图中 $\alpha=30^\circ$, $\beta=60^\circ$, $R=20\text{ cm}$ 为驱动轮中心到车体中心的距离。由图 1 可得机器人的速度分解公式为:

$$\begin{cases} v_1 = (V_x \sin \beta + V_y \cos \beta + V_r R) \times k \\ v_2 = (-V_x \cos \alpha + V_y \sin \alpha + V_r R) \times k \\ v_3 = (-V_x \cos \frac{\pi}{2} - V_y \sin \frac{\pi}{2} + V_r R) \times k \end{cases} \quad (1)$$

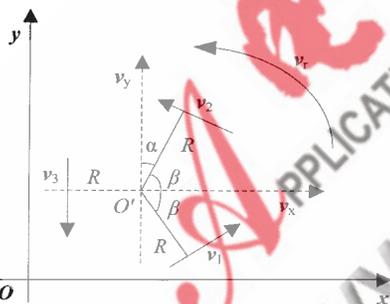


图 1 机器人坐标系及速度分解示意图

其中, 速度比例系数 $k=0.035$, 线速度单位为 m/s , 角速度单位为 rad/s 。

2.2 三维物体建模

ODE 物理引擎提供球体、盒子、胶囊、平面和圆柱等几何体。在创建一个几何体后, 其中心一般落在仿真环境坐标系的原点上。在对移动机器人进行建模时, 需要使用盒子和圆柱两种几何体, 盒子需要指定 3 个参数, 即长、高、宽; 对于圆柱而言, 则需指定长度和半径 2 个参数。在动力学世界中, 以 Geom 代表物体几何体, 以 Body 代表虚拟场景中的对象。软件可以通过调用 ODE 内部函数来检测几何体和对象的对应关系, 也有函数用于检测对象之间是否存在连接。下面以创建盒子物体为例, 说明单个物体的建模方法。

《微型机与应用》2011 年 第 30 卷 第 4 期

首先调用 `dBodyCreate` 函数创建出给定空间中的刚体对象, 再调用 `dBodySetPosition` 和 `dBodySetRotation` 两个函数, 调整该对象在空间中的位姿, 接着调用 `dMassSetBoxTotal` 和 `dBodySetMass` 两个函数设定该对象的质量属性, 最后调用 `dCreateBox` 函数创建相应尺寸盒子几何体, 并调用 `dGeomSetBody` 函数将该几何体与对象关联起来。

在创建出单个对象后, 往往需要利用各种关节将不同对象连接起来。ODE 物理引擎提供 5 种类型的关节, 分别为铰链型、球-球窝型、滑竿柱型、固定型和铰链 2 型等。其中铰链型为合页关节, 滑竿柱型为插销关节, 铰链 2 型则是带有轴的关节, 这些关节都有内置的马达。本文选用铰链关节来连接驱动轮和机器人车体, 采用固定关节构建机器人车体结构。下面以使用铰链关节连接两个对象为例, 说明创建关节的方法。

在调用 `dJointCreateHinge` 函数创建铰链关节对象后, 再调用 `dJointAttach` 函数指定用该关节连接的两个物体对象, 然后调用 `dJointSetHingeAnchor` 函数设定旋转轴的中心点坐标, 并调用 `dJointSetHingeAxis` 设定旋转轴的方向。

为了让刚体对象能够在仿真环境中运动起来, ODE 提供了 3 种方法: (1) 调用 `dBodyAddForce`, `dBodyAddTorque` 等函数给刚体施加力的作用; (2) 调用 `dJointSetHingeParam` 函数来改变内置马达的转速, 同时需指定该函数的第二个输入参数为 `dParamVel`; (3) 调用 `dBodySetLinearVel` 和 `dBodySetAngularVel` 两个函数直接给物体设定线速度和角速度。

此外, 在 ODE 仿真环境中, 可通过两种方式来模拟弹簧-阻尼系统: (1) 通过设置 ERP (Error Reduction Parameter) 和 CFM (Constraint Force Mixing) 两个参数来实现, ERP 为每一仿真循环中的修正误差, 取值范围为 $0 \sim 0.8$, 默认取值为 0.2 ; CFM 代表物理引擎的全局混合约束力, 它反映物体表面的柔软程度, 其取值范围为 $10e^{-9} \sim 1$; (2) 利用动力学方程来求解, 即胡克定律:

$$F = -kx \quad (2)$$

其中 k 为物体的劲度系数, x 为物体形变量。考虑阻尼系数时, 动力学方程为:

$$m \frac{d^2x}{dt^2} = -kx - \gamma \frac{dx}{dt} \quad (3)$$

其中 γ 为阻尼系数, 它与物体的形状以及周围性质有关。

综上所述, 典型 ODE 仿真过程为^[6]:

- (1) 生成一个动力学世界, 并在该世界中创建物体;
- (2) 设置物体状态 (如质量、质心位置和姿态等), 并在动力学世界中创建关节;
- (3) 将关节与物体绑定起来, 为所有的关节设置参数;
- (4) 生成碰撞检测空间, 并为需要作碰撞检测的物体生成碰撞几何体; 创建一个容纳关节的关节组;
- (5) 循环处理过程:
 - ① 在物体上施加力;

欢迎网上投稿 www.pcachina.com

- ② 根据需要调整关节参数;
- ③ 调用碰撞检测, 得到碰撞点和碰撞的物体;
- ④ 为每个碰撞点的碰撞创建一个接触关节, 并将其放入关节组;
- ⑤ 执行一个仿真步骤;
- ⑥ 清空接触关节组中的关节;
- (6) 销毁动力学世界和碰撞世界。

2.3 三维图形的绘制

OpenGL 绘制图形的基本操作步骤^[4]:

- (1) 设置像素格式: 设定 OpenGL 绘制风格、颜色模式和颜色位数等重要信息。
- (2) 建立模型: 根据基本图元建立景物的三维模型, 并对模型进行数学描述。
- (3) 舞台布置: 把景物放置在三维空间的适当位置, 设置视点、视角和投影模型等。
- (4) 效果处理: 设置物体的材质, 加入光照及光照条件。
- (5) 光栅化: 把景物及其颜色信息转化为可在计算机屏幕上显示的像素信息。

在绘制图形时, 需注意坐标系的变换, 否则很容易导致绘制失败。OpenGL 定义了两个坐标系: 世界坐标系和当前绘图坐标系。世界坐标系是固定不变的, 规定以屏幕中心为原点, 面对显示终端, 向右为 x 正轴, 向上为 y 正轴, 向终端外面为 z 正轴。当前绘图坐标系是绘制物体时的参考坐标系。仿真软件完成初始化后, 世界坐标系与当前绘图坐标系是重合的。在调用 $glTranslatef$ 、 $glRotate$ 等变换函数对绘图坐标系进行平移和旋转后, 绘图坐标系会在原来的基础上做出相应改变。此时, 若调用 $gluSphere$ 、 $glVertex3f$ 等绘图函数, 绘图函数是在改变之后的绘图坐标系上进行绘制。如若要让绘图坐标系与世界坐标系再次重合, 可以调用 $glLoadIdentity$ 函数。此外, 可以调用 $glColor3f(r, g, b)$ 函数设置绘图函数所使用的颜色, 如果没有再次调用该函数, 则绘制出的图形颜色将保持原先颜色不变, rgb 三个颜色分量的取值范围为 $0.0 \sim 1.0$ 。

3 软件框架

软件框架及其处理流程如图 2 所示, 软件仿真循环的处理流程如图 3 所示, 下面简述主要处理过程的实现思路及方法:

(1) 软件的初始化

软件的初始化工作主要包括: 从配置文件读入移动机器人的数量、软件运行时所需文件的路径和绘制窗口大小等参数; 导入虚拟场景所需的素材, 如声音、纹理和 3ds 模型等, 本文采用 lib3ds 库来读取 3ds 文件^[6]; 初始化 ODBC 接口, 获取 ODBC 环境句柄, 实现进程与数据库的数据交互; 初始化游戏手柄和 UDP 通信网络; 初始化 ODE 环境, 创建动力学环境和碰撞检测空间, 创建移

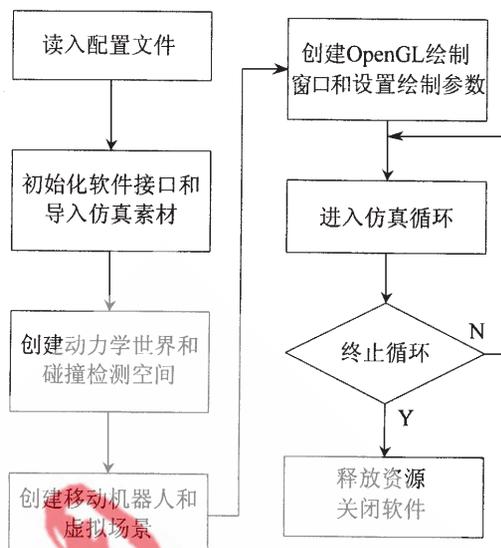


图 2 软件框架及处理流程



图 3 仿真循环的处理流程

动机器人和仿真场景。

(2) 物体运动参数的接收

在仿真循环中, 利用 UDP 协议实现本地进程与其他进程之间的通信, 该软件也可以从键盘、游戏手柄和网络数据库读入机器人的动作指令。

(3) 物体碰撞检测

本文将与物理引擎有关的操作函数封装成动态链接库, 通过使用动态链接库技术, 软件可以实现模块化, 即由相对独立的组件来组成整个软件。这简化了软件项目的管理, 而且能够节省内存, 也有助于资源共享和代码更新移植。

ODE 的碰撞检测引擎需要给定两个物体的形状信息。在每一仿真循环中, 调用 $dSpaceCollide$ 函数获取可能发生接触的物体, 再由该函数指定的碰撞回调函数 $nearCallback$ 将接触点信息传给用户。由此, 用户可根据自身需要建立物体间的碰撞连接点, 每个连接点都有相

《微型机与应用》2011 年第 30 卷 第 4 期

应的 dContactGeom 结构,该结构保存着碰撞点的位置和两个物体互相进入对方的深度。为了提高仿真速度,软件使用较少的关节和较少的接触面,如果条件允许,也可以使用无摩擦或者粘性接触面。下面给出碰撞回调函数中关于接触面的参数设置:

```
surface.mu=3.0;           //库仑摩擦力系数
surface.mu2=0.0;         //库仑摩擦力系数 2
surface.slip1=0.05;      //摩擦力 1 方向的滑动摩擦力
surface.slip2=0.05;      //摩擦力 2 方向的滑动摩擦力
surface.bounce=0.9;      //反弹系数
surface.bounce_vel=1.0;  //反弹所需要的最小碰撞速度
surface.soft_erp=0.2;    //接触点法线方向“柔软”参数
surface.soft_cfm=1e-4;   //接触点法线方向“柔软”参数
```

(4) 三维图形绘制

在仿真软件中,与绘制图形有关的操作函数都被封装成动态链接库,该动态链接库只向仿真软件提供若干个接口,如在绘制机器人时,只需根据机器人的不同部位,设定相应的绘制参数,然后轮流调用 dsDrawCylinder、dsDrawBox 两个函数即可实现,dsDrawCylinder 函数的输入参数分别为物体位置、朝向、长度和半径,而 dsDrawBox 函数的输入参数为位置、朝向和尺寸;在绘制从 x 文件和 3ds 文件导入的三维模型时,只需调用 dsDrawTriangleD 函数即可。

4 仿真实例

本文在 VS2008 平台上开发仿真软件,ODE 版本为 0.9,OpenGL 版本为 1.0,软件为控制台程序。

在检验路径规划算法或避障算法时,可先创建一个虚拟足球场,再设定机器人初始位置的绝对坐标为(-9.0, -5.0),目的地绝对坐标为(5.5, 5.0),并在路径的中间布置 8 个或更多的障碍物,障碍物直径为 50 cm,其位置参数可以人为指定,也可以随机产生,然后在虚拟场景的二维地图上,以宏观鸟瞰的远程视野,对避障算法的仿

真结果进行观察、比较和分析,仿真效果如图4所示。若需了解算法运行的细节,可以直接观察3D环境的仿真过程,或通过分析记录下的仿真数据来比较算法的优劣。而在检验多机器人协作算法时,可同时利用三维环境和二维全局地图来对算法性能进行比较,以由多个移动机器人组成的足球队为例,移动机器人需按照协作算法来实现站位^[7]。

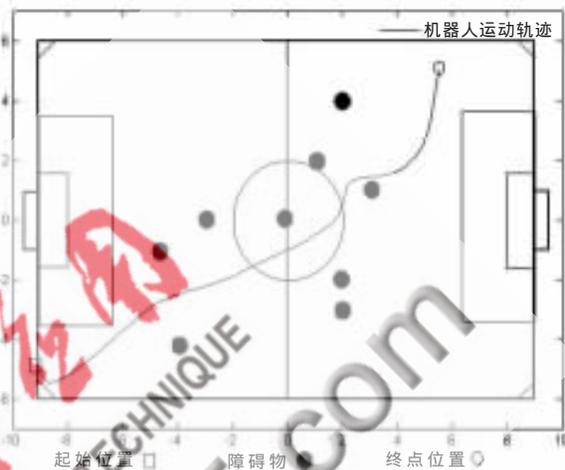
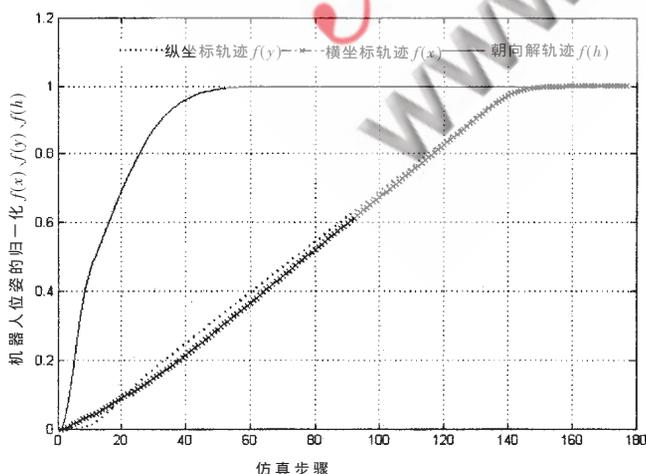
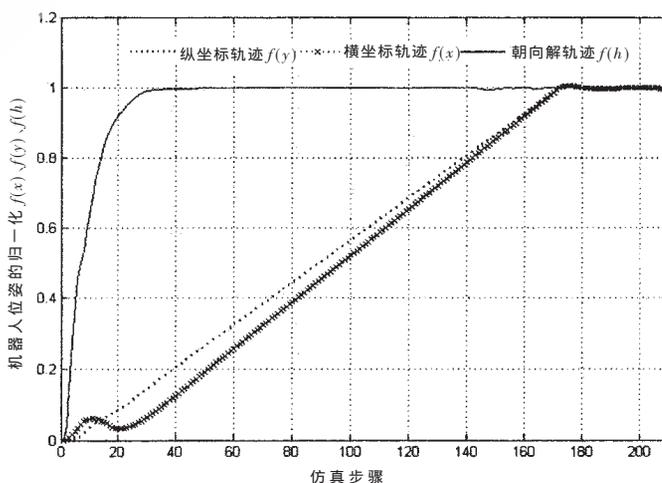


图4 在仿真平台上检验避障算法

此外,在采用PID控制算法对机器人位置进行控制时,有比例增益、微分增益和积分增益三个参数需要整定测试^[8],此时,可以利用仿真软件对参数进行步估计。实验方法是:先设定初始位置和目的地位置,然后让机器人以最快速度向目的地移动,当到达目的地后,就停止不动。在机器人移动过程中,记录下机器人在每一仿真步骤中的位姿 $(x,y,\theta)^T$,并将位姿数据归一化,以便能够将三维的向量绘制在同一坐标系下,同时记录机器人走完设定路径所需的时间。图5(a)为采用比例控制算法的机器人位姿变化轨迹,机器人走完该路径所需时间为16.239s,图5(b)为采用比例-微分控制算法的变化轨迹,所需时



(a) 采用比例控制的输出结果



(b) 采用比例-微分控制的输出结果

图5 采用不同控制算法的机器人位姿变化轨迹

间为 14.026 s。从图 5 的结果可看出,采用比例控制算法的系统输出无超调,而采用比例-微分控制算法的系统,其控制目标会出现微小超调,但机器人的响应速度明显提高,尤其是机器人朝向角的控制,能够更加快速准确地跟随给定值。

在移动机器人控制技术和多机器人协作技术的研究中,为了能够对算法参数进行有效检验和测试,本文利用 ODE、OpenGL 和 VS2008 开发出移动机器人仿真软件。该软件采用 ODE 生成动力学世界和模拟物体碰撞,充分利用了 ODE 的快速性和精确性,仿真软件还采用高效的图形接口 OpenGL 来绘制图形,提高软件的图形处理能力,改善图形显示效果。仿真实例证明,该软件具有较好的扩展性和实用性。

参考文献

- [1] 胡宇慧. 3D 游戏引擎中若干关键技术及算法的研究与分析[D]. 上海:上海师范大学, 2006.
- [2] 高高. 基于 OGRE 和 ODE 的驾驶模拟系统的设计与实现[D]. 武汉:武汉理工大学, 2006.
- [3] 欧阳慧琴, 陈福民. 物理引擎与图形渲染引擎绑定的研

究与实现[J]. 计算机工程与设计, 2008,29(21):5580-5620.

- [4] 蒋东翔. 基于 OpenGL 的 3D 网络游戏开发中关键技术的研究[D]. 哈尔滨:哈尔滨工程大学, 2008.
- [5] TAKEMURA Y. Development of "Hibikino-Musashi" omnidirectional mobile robot[J/OL]. http://robocup.ksrp.or.jp/hibikino-musashi/wp-content/uploads/2010/05/BrainIT2006_Ronbun.pdf.
- [6] DEMURA K. 3DS Loader for open dynamics engine-0.0 [CP/OL]. <http://demura.net/9ode/460.html>.
- [7] 邢关生, 张建勋. 移动机器人网络运动协调的控制与仿真[J]. 系统仿真学报, 2010,22(03).
- [8] 杨智, 朱海锋, 黄以华. PID 控制器设计与参数整定方法综述[J]. 化工自动化及仪表, 2005,32(05):1-7.

(收稿日期:2010-12-31)

作者简介:

刘润丹,男,1981年生,助教,硕士,主要研究方向:智能控制技术、机器人技术和图像处理。