

基于 OpenGL 的工业化机器人仿真研究*

董天平^{1,2}, 马燕¹, 刘伦鹏³

(1. 重庆师范大学 计算机与信息科学学院, 重庆 401331; 2. 重庆通信学院 基础实验中心, 重庆 400035;
3. 重庆师范大学, 物理与电子工程学院, 重庆 401331)

摘要:以工业机器人仿真系统为背景,介绍了机器人的发展历程,重点描述了工业化机器人仿真开发的方法。该方法掌握较为容易,开发成本低,具有良好的可扩展性。在此基础上,文章设计并实现了基于 OpenGL 的工业机器人仿真系统,包括虚拟仿真环境的搭建、运动学分析和三维模型的绘制。最后,研究了 OpenGL 在机器人图形仿真系统中的具体应用,通过代码封装使得工业机器人图形建模变得简易可行,并分析了工业机器人系统结构。探讨了该系统的运动学和逆运动学实现方法,取得了良好的效果。该仿真系统的研究方法也能够为其他场合的应用提供一定的参考价值。

关键词:工业机器人;图形仿真;OpenGL

中图分类号:TP242

文献标志码:A

文章编号:1672-6693(2013)02-0060-05

随着机器人技术的发展,机器人应用已经广泛渗透到国民各个经济领域。机器人已成为人类工业生产中不可缺少的好帮手,而且机器人的发展水平象征了一个国家的科技水平和经济实力,受到了众多国家的高度重视。在机器人的研究与设计中,图形仿真系统是一种安全灵活的工具,它发挥着非常重要的作用,并广泛应用于机器人研究和开发的各个方面^[1]。例如:能够为机器人整体的结构方案与设计提供有价值的参考依据,能够有效地帮助研究人员了解机器人工作空间的形态和极限;能够提供一种有效的验证手段给离线编程人员;还能够帮助检测作业环境与机器人之间的碰撞与干涉,以保证每个生产单元的绝对安全等。除了应用于机器人本身之外,图形仿真技术还能够根据特定作业帮助用户选择合适的机器人产品类型。

开发出一个有效的机器人图形仿真系统需要有相当强大高效的开发工具。在目前众多的开发工具中,OpenGL 是相对稳定而最具效率的一种^[2-3]。

1 工业机器人的仿真研究

在人工智能技术的高速发展的背景下,机器人在技术层次和功能上都有了很大的改善和提高。其中,移动机器人^[4]和机器人的视觉和触觉^[5-6]等技术就是最为典型的代表。这些技术的发展同样推动了机器人

概念的延伸。机器人已经日益广泛应用于社会的各个领域,而应用最多的则是工业生产中的机器人。工业机器人从60年代初开始为人们所用,经过了近50年的发展,现在已广泛应用于工业领域。根据目前工业焊接机器人在工厂快速发展的需要,尤其是汽车行业的大量应用,使得焊接过程的完全自动化成为人们研究的重要课题。当前,各国各大汽车厂商都已经引入机器人焊接生产线,通过降低生产成本提高生产效率来提升自己的核心竞争力。同时,将机器人的技术(如传感技术、智能技术、控制技术等)扩散和渗透到各个领域形成了各式各样的智能化机器人。

在机器人的仿真研究中,一般以 Windows 作为开发平台。在机器人的几何模型建立与仿真程序设计的过程中,还需要用到面向对象的程序设计方法^[7]。在设计机器人仿真模型时,目标模型是通过面向对象的详细、具体的分析过程而得出的,它为机器人的仿真、建立求解空间模型提供了很好的基础条件。在此,选用面向对象程序设计语言 Visual C++6.0 和 OpenGL 仿真编程作为开发工具,来完成机器人模型的设计和建立。Visual C++6.0 紧密结合了面向对象的3大思想:面向对象分析(OOA)、面向对象设计(OOD)和面向对象编程(OOP)。相比于传统的面向过程的编程方法,使用 Visual C++6.0 来开发大型的复杂的面向对

* 收稿日期 2012-05-28 修回日期 2012-08-26 网络出版时间 2013-03-16 13:37

资助项目:重庆高等教育教学改革项目(No. 102118);重庆市教委项目(No. KJ090809)

作者简介:董天平,男,硕士研究生,研究方向为人工智能和语义网络,E-mail:dongtianpingdtp@126.com;通讯作者:马燕,E-mail:mayan@cqnu.edu.cn

cqnu.edu.cn

网络出版地址: http://www.cnki.net/kcms/detail/50.1165.N.20130316.1337.201302.60_015.html

象的应用程序则更加方便有效。机器人设计仿真系统的原理如图 1 所示。

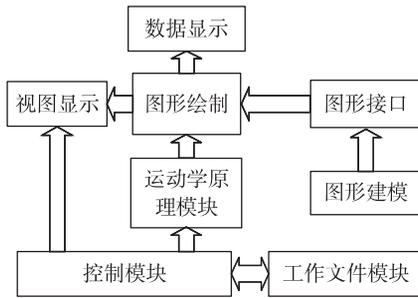


图 1 机器人仿真系统原理图

2 机器人仿真系统的设计

2.1 虚拟仿真环境建立

系统仿真一般都包括了 3 个基本的要素: 仿真模型、仿真系统和计算机。这 3 个基本要素的基本活动联系有: 模型仿真、系统建模和仿真试验。它们之间的关系如图 2 所示。

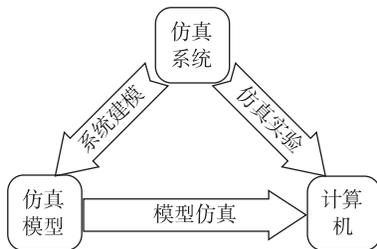


图 2 仿真系统 3 要素及 3 项基本活动

OpenGL 提供了一个具有良好兼容性的三维图形标准给程序开发人员。开发者常常使用 OpenGL 图像库来搭建仿真系统中的机器人三维视觉模型。它的开发目的是为了能够使用户专注于所做工作的本身, 只需要按照基本的图形影像常识进行建模、光照、渲染、布景等操作。开发者可以忽略开发工具的内容和形式, 更不需要考虑具体的开发软硬件条件。

由于 OpenGL 是一个纯应用程序编程接口(API), 所以必须搭建基本的应用程序框架来使用它的各种功能^[8]。

1) 设置 OpenGL 基础库。首先, 在 VC++ 的安装环境中导入 OpenGL 库, 其中, 需要配置的文件包括 LIB 文件、DLL 文件、H 头文件等。接着, 利用 VC++ 提供的 MFC 类库创建一个单文档类型的可执行程序, 建立相应的工程环境。最后, 在程序的 View.h 文件中写入下面的代码, 用来完成 OpenGL 头文件的添加。

```
#include "gl\gl.h"
#include "glu\glu.h"
```

在 VC++ 的设定选项中将程序的运行方式设定

为“静态库运行方式”, 通过这种方式可以使程序在没有安装 VC++ 的计算机上也能够顺利地运行。最后, 还要添加两个 OpenGL 的库文件(OpenGL32.lib, glu32.lib)到 VC++ 的链接选项卡中。

2) 添加变量、函数和消息响应。需要将成员变量、函数和消息响应添加到 View 类中, 以实现程序功能。

① 添加变量到 View 类中。

```
CRect my_oldRect;
CClientDC * my_pDC;
```

② 添加成员函数到 View 类中。

```
BOOL MyPixelFormat() // 描述象素格式
void MyFirst() // 渲染描述表
void MyDraw() // 绘制图形
```

③ 添加消息响应到 View 类中。

```
ON_WM_CREATE();
ON_WM_SIZE();
ON_WM_DESTROY();
```

3) 设置象素的格式。在 Windows 环境中, 应用程序一般通过使用设备描述表(Device context)来实现图形的绘制和显示。然而, OpenGL 是通过使用渲染描述表(Rendering context)来实现对图形的操作。该描述表的关键部分是实行对象素格式的设置。因此, 当通过 OpenGL 来建立模型的时候, 实际上是对设备象素的操作, 而不是简单的绘图操作。OpenGL 先将数据转化成象素操作, 再转送到帧缓存中, 这时执行程序就需要知道 Windows 的象素格式。不然, 程序就无法完成对设备象素的操作和使用。为了解决此问题, OpenGL 通过定义如下象素格式结构体 PixelFormatDescriptor 来实现象素格式的设置, 其中包括颜色模式定义、颜色位数设置、缓存设置等。

```
static PixelFormatDescriptor pfd = { sizeof( PixelFormatDescriptor ) // 描述结构体的大小
PFD_Type_RGBA // 定义为 RGBA 颜色模式
32 // 定义为 32 位颜色
PFD_DoubleBuffer // 描述应用缓冲模式
...
};
```

4) 使用渲染描述表。

① 在成员函数 MyFirst() 中创建渲染描述表(RC)。hrc = wglCreateContext(my_pDC -> GetSafeHdc()) // 创建描述表

```
wglMakeCurrent( my_pDC -> GetSafeHdc(), hrc ) // 渲染当前描述表
```

② 调用渲染描述表(RC)。在消息响应函数 ON_WM_CREATE() 中, 通过调用成员函数 MyFirst() 来调

用渲染描述表。

5)设置窗口背景。在 OnDraw()函数中完成对窗口背景的设置。

```
glClearColor( 0 0 0 0 ) //设置窗口背景的颜色
```

```
MyDraw( ) //绘制图形
```

```
GLFinish( ) //完成图形绘制
```

6)设置视点和视场。

①在成员函数 MyFirst()中完成视点和视场的设置。

```
GLMatrixMode( GL_ModelView ) //启动模型矩阵
```

```
GetClientRect( &my_oldRect ) //描述用户窗口的尺寸
```

②在消息响应函数 ON_WM_SIZE()中完成视场的变化。

经过以上步骤的设置,开发程序的基本框架就已经搭建完成,开发者可以在 View 类中的成员函数 MyDraw()中进行图形的绘制了。为了方便使用 OpenGL 程序进行开发,还可以利用 VC++ 提供的定制功能。它可以将搭建好的框架设定成开发项目选项。这样就使得开发者不必每次都重复以上的设置步骤,从而大幅度提高程序开发的效率。

2.2 机器人结构和运动学分析

2.2.1 工业机器人系统结构分析

工业机器人系统的全局可靠度完全依赖于各个分配单元的可靠度。在建立模型的过程中,假定工业机器人系统各单元的可靠度为 R_1, R_2, \dots, R_n , 且 $R_1 \leq R_2 \leq \dots \leq R_n$, 则可以计算出整个工业机器人系统可靠度值为

$$R_s = \prod_{i=1}^n R_i \quad (1)$$

在该系统中,由于系统总可靠度等于其组成单元的可靠度之积,同理可得,每个单元的可靠度等于各自组成元件的可靠度之积。因此可以得出结论:如果要大幅度提高系统的整体可靠度,则提高某一单元的可靠度的效果好于提高此单元中某一组成元部件可靠度的效果,而且,提高低可靠度单元的可靠度的效果要大大好于提高低可靠度元件的可靠度。

2.2.2 工业机器人运动学模拟

本文设计的模型如图3所示,通常采用机座坐标系 $O-XYZ$ 来描述工业机器人夹持器的位置和姿态。三维场景可以环绕世界坐标系的 X, Y, Z 轴自由旋转。由于世界坐标系是固定不动的,它不因为三维场景的旋转而改变,因此在这里需要构造变换矩阵 T , 记录下该三维场景每次旋转的累积效果。然后再通过函数 `GLMultMatrixd(const GL double * m)` 将变换矩阵 T 与当前模型视图矩阵相乘,以此来实现整个场景旋转^[9]。

本文所设计的工业机器人各连杆坐标系如图3所示。

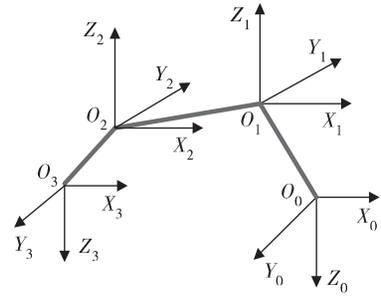


图3 机器人各连杆坐标系

则该机器人的各个变换矩阵分别为

$${}^0_1T = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^1_2T = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S_2 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^2_3T = \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^0_2T = {}^0_1T * {}^1_2T = \begin{bmatrix} C_1C_2 & -C_1S_2 & -S_1 & 0 \\ S_1C_2 & -S_1S_2 & C_1 & 0 \\ -S_2 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$${}^0_3T = {}^0_1T * {}^1_2T * {}^2_3T = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & -S_1 & C_1C_2 \\ S_1C_{23} & -S_1S_{23} & C_1 & S_1C_2 \\ -S_{23} & -C_{23} & 0 & S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

其中 $C_1 = \cos \theta_1, S_1 = \sin \theta_1, C_{23} = C_2C_3 - S_2S_3, S_{23} = C_2S_3 + S_2C_3$ 。

2.3 工业机器人逆运动学及三维模型的绘制

2.2节的内容初步描述了工业机械臂的正运动学理论知识。在实际的工业机器人模型的建立过程中,正运动学和逆运动学是同时存在互为补充的,这两个理论共同支撑起机器人运动学的基础。若已知机械臂末端执行器的坐标位置,则可以通过该位置坐标计算出机械臂各个关节的角度值。将以上求解关节的角度值的过程称作求解逆运动学问题,它是正向运动学问题的一个反过程。求解正向运动学问题,方法相对简单且结果唯一确定;而求解逆向运动学问题则相对复杂,它有可能存在很多种解,也可能由于条件约束而不

存在解。在工业机器人存在多种运动学反解的情况下,应该根据具体的目标条件来选择其中的一组最优解,例如:满足功率最省、行程最短、回避障碍最佳、受力最好等要求。

由于机器人逆运动学问题的求解存在多种方法,各种方法都有自身的优点和不足之处。本文主要通过反变换法来求解工业机器臂的运动学反解。

一般情况下,给定运动方程

$${}^0_3T = {}^0_1\pi(\theta_1) * {}^1_2\pi(\theta_2) * {}^2_3\pi(\theta_3) \quad (7)$$

要求得 θ_1 , θ_2 和 θ_3 , 一般的求解过程为

$$\left. \begin{aligned} {}^0_1T^{-1}(\theta_1) {}^0_3T &= {}^1_2\pi(\theta_2) * {}^2_3\pi(\theta_3) \Rightarrow \theta_1 \\ {}^0_2T^{-1}(\theta_1, \theta_2) {}^0_3T &= {}^2_3\pi(\theta_3) \Rightarrow \theta_2, \theta_3 \end{aligned} \right\} \quad (8)$$

经过机器人运动学分析,可以得到在建立工业焊接机器人模型的过程中所需的位置参数和各个关节的角度参数,从而能够方便地绘制出焊接机器人的基本模型。通过机器人动力学模型分析,进一步为机器臂的轨迹优化奠定基础。

通过利用 OpenGL 强大的图形绘制功能来建立工业机器人的基本模型。在 View 类的成员函数 MyDraw() 中完成虚拟的工作空间场景和机器人图像模型的绘制。由于 OpenGL 为开发者提供了大量的有关基本图形的绘制程序,所以作者可以非常方便地通过这些函数来完成对机器人的三维建模^[10]。

作者利用 OpenGL 程序绘制机器人的运动效果图,它是根据规划的运动轨迹来实现图形的更新显示。程序首先需要对规划轨迹的信息点进行确定,然后求得每一个轨迹插值点的关节角度,最后通过利用图形程序接口来完成窗口的图像更新显示,以此来达到动态显示动画图像的效果。整个动画图像的更新显示过程流程如图 4 所示。

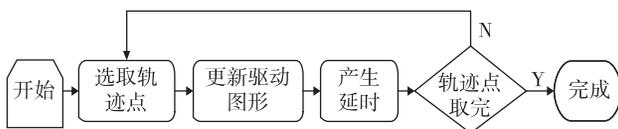


图 4 动画图像显示流程图

3 OpenGL 在机器人图形仿真中的具体应用

OpenGL 提供的强大的三维图形功能使这个工业机器人图形建模变得简易可行。利用 OpenGL 开发的工业机器人简单模型如图 5 所示。

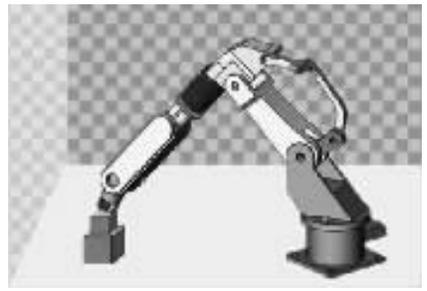


图 5 工业机器人模型仿真结果

与 Windows 下提供的图形设备接口(Graphical device interface, GDI)相比较,OpenGL 具有十分显著的优点。它可以设置三维模型的材质和光照,并且能够通过使用函数直接构造三维多面体并设置三维多面体的投影参数。因此,在使用 OpenGL 开发模块时,通常能够直接调用其自身提供的函数来实现对三维图形的操作。只需要提供有关三维多面体在实际三维空间中的坐标参数给 OpenGL 程序,它就可以绘制出有表面材质的、带光照的三维多面体来,从而使工业机器人模型达到形象、逼真的效果。OpenGL 不需要用户自己设计和编写投影算法,来计算三维模型的每一个顶点在二维平面上的投影,然后画出点与点之间的连线。从选择的投影方式来讲,OpenGL 还为开发者提供了多种投影方式的设置函数,使用户能够方便快捷地选择平行投影或透视投影,并具体场景来设置投影角度。OpenGL 还提供了使三维图形实体能够围绕工作空间中的任何一条直线进行旋转或者沿着该直线进行平移的变换函数,所以用户能够自由地从不同的视距和角度对工业机器人模型的状态进行观测调整。此外,当考虑到画面实时性以及平滑的动画效果的时候,计算机需要充分利用到内存的缓存机制,它需要先在内存中生成下一幅将要显示的图像,然后将已经生成的图像从内存中发送到的屏幕的显示空间中。在这个过程中,OpenGL 的设备无关位图(Device independent bitmap, DIB)技术和双缓存技术(Double buffer, DB)均发挥着巨大的作用。在实际机器人仿真编程开发中,作者构造诸多 OpenGL 类,并将这些代码封装起来以供后续开发程序使用。

4 结论

随着对工业机器人研究的不断深入,使得工业机器人自动化控制的应用范围越来越广。由此带来的经济效益和社会影响也变得越来越显著。基于 OpenGL 的仿真研究方法能够使开发者容易掌握。在工业应用中,它可以大幅度降低开发成本,加快开发速度。该方法具有良好的可扩展性,并且能够为其他场合的应用

提供一定的参考价值。值得说明的是,在对于相对大型的高质量、高分辨率的图形工作时,该系统需要在软硬件的密切配合中才能将显示效果更好地发挥出来。本文提出的基于 OpenGL 的工业机器人仿真应用,理论证明清晰明确,在实际的应用中,取得了比较理想的效果。

参考文献:

- [1] 陈伟,陈一民,汪地,等. 机器人虚拟仿真及控制技术研究 [J]. 计算机工程与设计 2010, 31(20): 4455-4458, 4496.
Chen W, Chen Y M, Wang D, et al. Research on virtual simulation and control of robot [J]. Computer Engineering and Design 2010, 31(20): 4455-4458, 4496.
- [2] 刘芳,刘贤梅. 3DS 文件读取、绘制与控制方法的研究与应用 [J]. 计算机工程与设计 2009, 30(19): 4575-4578.
Liu F, Liu X M. Research and application on method of 3DS file read, render and control [J]. Computer Engineering and Design 2009, 30(19): 4575-4578.
- [3] 王俊骅,张方方,张兰芳. 基于 OpenCV 和 Halcon 的交通冲突视频自动检测及数据处理 [J]. 同济大学学报:自然科学版 2010, 38(2): 238-244.
Wang J H, Zhang F F, Zhang L F. Halcon and openCV-based traffic automatic conflicting detecting method and data transaction [J]. Journal of Tongji University: Natural Science 2010, 38(2): 238-244.
- [4] 赵娟平,高宪文,符秀辉. 改进蚁群优化算法求解移动机器人路径规划问题 [J]. 南京理工大学学报, 2011, 35(5): 637-641.
Zhao J P, Gao X W, Fu X H. Improved ant colony optimization algorithm for solving path planning problem of mobile robot [J]. Journal of Nanjing University of Science and Technology, 2011, 35(5): 637-641.
- [5] 王一,刘常杰,杨学友,等. 工业机器人视觉测量系统的在线

校准技术 [J]. 机器人 2011, 33(3): 299-302, 331.

Wang Y, Liu C J, Yang X Y, et al. Online calibration of visual measurement system based on industrial robot [J]. Robot, 2011, 33(3): 299-302, 331.

- [6] 秦岚,裴利强. 新型智能机器人触觉传感服装阵列标定的研究 [J]. 传感器与微系统 2011, 30(7): 71-74.
Qin L, Pei L Q. Calibration research on a new type of tactile sensing suit array for intelligent robot [J]. Transducer and Microsystem Technologies 2011, 30(7): 71-74.
- [7] 王小兵,段振华. 框架时序逻辑语言 MSVL 中面向对象机制的实现 [J]. 西安电子科技大学学报, 2010, 37(3): 559-564, 575.
Wang X B, Duan Z H. Implementation of objected-oriented mechanism in framed temporal logic language MSVL [J]. Journal of Xidian University 2010, 37(3): 559-575.
- [8] Shreiner D, Woo M, Neider J, 等. OpenGL 编程指南 [M]. 徐波译. 北京:机械工业出版社, 2006.
Shreiner D, Woo M, Neider J, et al. OpenGL programming guide [M]. fifth edition. Beijing: China Machine Press, 2006.
- [9] 张慧勇,赵曜,兰杰. 足球机器人的数学模型及最优控制方法 [J]. 云南大学学报:自然科学版 2009, 31(S2): 187-191.
Zhang H Y, Zhao Y, Lan J. The mathematical model of soccer robot and optimal control [J]. Journal of Yunnan University: Natural Sciences Edition 2009, 31(S2): 187-191.
- [10] 王建中,朱礼尧. Three-Dimensional kinematics simulation of eobot fighting platform in virtual environment [J]. 北京理工大学学报:英文版 2010(S2): 165-169.
Wang J Z, Zhu L Y. Three-Dimensional kinematics simulation of robot fighting platform in virtual environment [J]. Journal of Beijing Institute of Technology: English Version, 2010(S2): 165-169.

Industrial Robot Simulation Research Based on OpenGL

DONG Tian-ping^{1, 2}, MA Yan¹, LIU Lun-peng³

(1. College of Computer and Information Science, Chongqing Normal University, Chongqing 401331;

2. Basic Experiment Center, Chongqing Communication Institute, Chongqing 400035;

3. College of Physics and Electronic Engineering, Chongqing Normal University, Chongqing 401331, China)

Abstract: In the background of simulation system and industrial robot, this paper described the development of the robot and focused on the simulation method of industrial robot. This method was easy to master, reduced development costs, and had good expansibility. Then, this paper designed and implemented industrial robot simulation system based on the OpenGL, which includes the construction of the virtual simulation environment, the analysis of kinematics and the drawing of 3D model. Finally, this paper studied the specific application of OpenGL in the robot graphics simulation system, discussed the system realization method, and achieved good results. The simulation system also had a certain reference value for the other applications.

Key words: industrial robot; graphic simulation; OpenGL