

基于 OpenGL 的六自由度机器人建模仿真

解伟, 王晓璨

(西北工业大学 自动化学院, 陕西 西安 710072)

摘要: 介绍了在交互式 C 语言开发平台 LabWindows/ CVI 下, 利用 OpenGL 图形系统的功能来设置 LabWindows/ CVI 与 OpenGL 的图形接口、建立 OpenGL 光照、视图和渲染描述表及利用 OpenGL 基本图元建立六自由度机器人的三维模型的新方法, 并且利用 OpenGL 的双缓存技术实现了仿真系统的交互式界面。在此开放的仿真系统下, 可直观、正确地模拟机器人的运动情况, 还能方便地进行机器人运动轨迹的仿真研究。

关键词: 机器人; 三维建模; 双缓存技术; 运动仿真

中图分类号: TP242.6 **文献标识码:** A

Modeling and simulation of 6-DOF manipulator based on OpenGL

XIE Wei, WANG Xiao Can

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: In this paper, a new method of configuring graphics interface of LabWindows/ CVI and OpenGL, and how to use a function of OpenGL graphics system to build lighting, view, rendering context and 3D simulation system of 6-DOF manipulator, is introduced under ANSI C language integrated development environment LabWindows/ CVI. And using the OpenGL double buffer animation technology to design interactive simulation system interface. It can exactly simulate the movement of manipulator under the simulation system, and can process simulation research of manipulator kinematics control.

Key words: manipulator; 3D modeling; double buffer technique; motion simulation

机器人三维建模仿真是机器人各项仿真中的一个重要组成部分, 对于机器人的运动轨迹仿真有重要的指导意义。LabWindows/ CVI(以下简称 CVI) 是美国 NI 公司推出的交互式 C 语言开发平台。其集成化开发环境、交互式编程方法、函数面板和丰富的库函数大大增强了语言的功能, 为熟悉 C 语言的开发人员提供了一个理想的软件开发平台^[1]。由于在 CVI 中已经集成了 OpenGL 图形标准, 并且 OpenGL 编程类似于 C 编程, 实际接口就是 C, 相对基于 Visual C++ 基础类库 MFC 和 OpenGL 开发的三维仿真平台, 在 CVI 中利用可视化技术, 开发基于 OpenGL 的三维图形更方便快捷, 同时可在此三维模型平台上方便地进行轨迹规划和控制算法等方面的仿真研究。

1 OpenGL 的简介和基本操作过程

1.1 OpenGL 简介

OpenGL 图形系统是图形硬件的一种软件接口。OpenGL 是 SGI 公司开发的开放式三维图形标准, 它实质上就是一个三维图形和模型库, 具有高度的可移植性, 可以在多种操作系统平台上运行, 例如各种版本的 Windows、Unix/ Linux 和 Mac OS 等, 具有非常快的渲染速度。它可以对计算机图形技术进行控制, 产生逼真的图像或者虚构出现实世界没有的图像, 是高性能图形和交互式场景处理的工业标准^[2]。OpenGL 是一种高性能的开放式图形库, 它的图形 API 以函数形式提供了 115 个核心库函数、43 个实用库函数 GLU、31 个编程辅助库函数 GLAUX 以及若干个 X-Windows/MS-Windows 专用库

函数 GLX/WGL。OpenGL 提供了多种三维图形的绘制方法，包括线框绘制方式、深度优先方式、反走样方式、平面明暗处理方式、光滑明暗处理方式、阴影和纹理方式、运动模糊方式、大气环境效果、深度域效果^[3]。OpenGL 的这些功能可以实现逼真的三维绘制效果，创建交互性三维场景和视景动画。

1.2 OpenGL 的基本操作过程

OpenGL 对三维图形操作可以归纳为：场景描述、设置光照、观察场景、光栅化，这个过程与人们观察世界的过程是一致的，其具体操作步骤如下：

(1) 根据基本图形建立景物模型，并对所建立的模型进行数学描述；

(2) 把景物模型放在三维空间中合适的位置，并设置视点 (Viewpoint)，以观察场景；

(3) 计算模型中所有物体的色彩，同时确定光照条件、纹理映射方式等；

(4) 把景物模型的数学描述及其色彩信息转化至屏幕上的像素，即光栅化 (Rasterization)。在这些步骤的执行过程中，OpenGL 还可能执行其他操作，如图 1 所示。

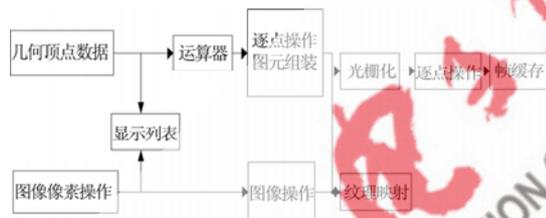


图 1 OpenGL 工作的基本流程图

程序从图 1 的左侧进入，经过一系列的运算处理，将几何顶点数据和图像像素数据加工后生成待显示的帧。另外，景物光栅化之后被送入帧缓存前，还可以根据需要对像素进行操作。

2 在 LabWindows/CVI 中配置 OpenGL

在 CVI 开发环境中使用 OpenGL 开发三维仿真图形的关键是如何配置 CVI 和 OpenGL 的图形接口。

2.1 添加与 OpenGL 相关的头文件和库文件

在 CVI 下创建一个新的工程文件：OGLHand.prj，在 OGLHand.c 文件首部添加 4 个头文件：

```
#include<windows.h>
#include<GL\gl.h> // 基本库
#include<GL\glu.h> // 实用库
#include<GL\glaux.h> // 辅助库
```

在 CVI 编辑器的主菜单“edit”中选取“Add Files to Project”下的“Library”，在弹出的对话框里选取 CVI\sd\lib

中的 3 个库文件：glu32.lib、glaux.lib 和 opengl32.lib 并添加到当前工程中。

2.2 设置 CVI 与 OpenGL 图形接口

虽然 OpenGL 已经内嵌在 CVI 中，但是要让 CVI 控件显示 OpenGL 图形，必须把 CVI 的 picture 控件转换成 OpenGL 控件。需要在工程中添加函数 cviogl.fp，其路径为：CVI\tools\lib\custctrl。并在 OGLHand.c 文件首部添加：
#include"cvigl.h"。

在主函数 main() 里调用函数将 CVI 的 picture 控件转换成 OpenGL 控件并返回控件句柄 OGLCtrl，再调用 InitControl() 函数初始化 OpenGL 属性，最后调用 int OGLRefreshGraph(int Panel_Handle, int OGL_Control_Id) 函数及时更新 OpenGL 控件图形的场景和图形属性。

3 基于 LabWindows/CVI 和 OpenGL 的机器人三维建模

3.1 建立 OpenGL 光照、视图及透视模式

视图决定模型在场景中的位置，同时要选择一个有利的观察点来观察场景。通过定义光照、视图及透视模式就能在二维屏幕上显示出立体的三维图形，这些都放在 InitControl() 函数中定义，包括初始化系统光照和初始化视图位置。

```
// 初始化系统光照
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_LIGHTING_ENABLE, 1);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_LIGHT_SELECT, 1);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_LIGHT_ENABLE, 1);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_LIGHT_DISTANCE, 2.0);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_LIGHT_LATITUDE, 30.0);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_LIGHT_LONGITUDE, 150.0);
// 初始化系统视图位置
OGLSetCtrlAttribute(hPanel, this->hControl,
    GLATTR_PROJECTION_TYPE, OGLVAL_PERSPECTIVE);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_VIEW_DIRECTION, OGLVAL_USER_DEFINED);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_VIEW_CENTERX, 0.0);
```

```
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_VIEW_CENTERY, 0.0);
OGLSetCtrlAttribute(hPanel, this->hControl,
    OGLATTR_VIEW_CENTERZ, 1.0);
```

3.2 建立渲染描述表

渲染图像就是计算机根据模型创建图像的过程，即三维图形的显示过程。要把一个物体的三维坐标变换为屏幕上的像素坐标，最终使图形在屏幕上显示。

OpenGL坐标变换包括：几何变换、投影变换、剪裁变换和视区变换，三维图形的显示流程如图2所示。

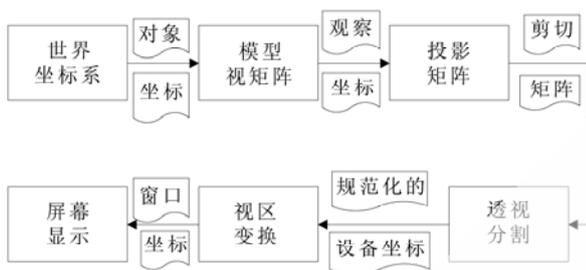


图2 三维视图显示流程

视区变换矩阵： $GLViewport(x,y,width,height)$ ，其中，参数 x,y 是视区在屏幕窗口坐标系中的左上角坐标， $width$ 和 $height$ 分别为视区的高度和宽度。注意变换只对其后的一系列操作产生影响，如在此之前屏幕上已有图形，则变换对其不产生影响。变换后如交换缓存，则可形成动画，否则只在平面上作图。

3.3 建立六自由度机器人模型

OpenGL 提供对二维和三维图形的基本操作，但是并不提供描述复杂几何物体及建立复杂物体模型的手段，因而必须寻找自身的三维建模方法。对不太复杂的模型直接利用 OpenGL 的基本三维图元（点、线、面）来构造，既方便又快捷，易于对模型进行操作和控制，缺点是建模的灵活度不大，不过对于建造本系统的三维模型已足够^[4]。

对实物进行一定简化后建立的模型是由一些圆柱体、长方体和球体组合而成。采用顺序描述各个子体的方法来建立六自由度机器人模型。 $gluCylinder()$ 用来绘制一个中空的二次柱面作为底座。 $gluDisk()$ 用来绘制一个垂直于 Z 轴的圆盘。 $gluSphere()$ 用来绘制一个球体作为关节。 $glRotatef()$ 表示把当前矩阵与一个表示旋转的物体的矩阵相乘，以逆时针的方向绕着从原点到目标点的直线旋转。参数 $Rotation$ 指定了旋转的度数。 $glTranslatef()$ 把当前矩阵与一个表示移动物体的矩阵相乘，这个移动矩阵由函数的参数指定，经过平移的矩阵成为当前

的变换矩阵。由于在 OpenGL 里描述下一个物体都是在前一个物体的基础上，因此每次都需要对新坐标中心进行定位。例如当创建好机器人手臂的第一段后，为了创建手臂的第二段，需要使用函数 $glTranslate()$ 把局部坐标系统移动到下一个节点。

模型的构建过程采用分层结构，即在相应的关节坐标系中构建各个模块，这样可以实现独立控制，使其能够绕着各自的坐标轴旋转。两只机器人手的三维图形如图3所示。

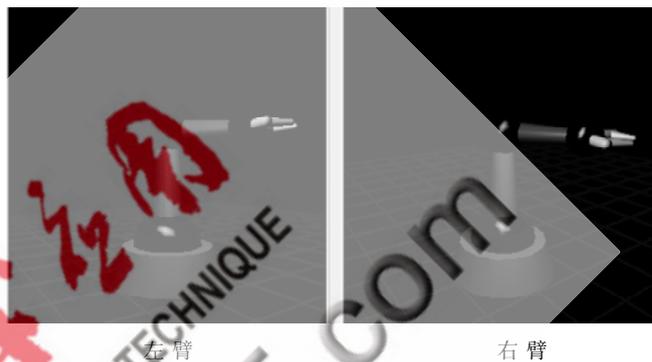


图3 机器人手三维模型

3.4 机器人可视化仿真

机器人仿真的研究已经成为机器人学中一个引人注目的领域，对于验证机器人工作原理、工作空间及进行运动学研究等都具有非常重要的指导意义。而机器人作为机器人不可缺少的部分，对其进行三维可视化仿真可以使研究者对机器人运动方式有一个感性的认识，可以帮助研究人员对机器人身体各个部分的位置关系建立正确的方程，可用于检验轨迹规划和作业规则的正确性和合理性，可为离线编程技术的研究提供一种有效的验证手段。显然，采用可视化仿真技术的好处在于：

(1) 从上位机的图形上就可以看到机器人手的运动情况，增加了视觉效果；

(2) 对于机器人危险环境或者远距离工作，而人又无法到达现场时，采用此项技术的意义是明显的，它可以保证人在上位机系统进行实时监控机器人手的工作状况，以便发出新的控制指令。

3.4.1 实现控制机器人仿真界面

在仿真界面上设置了机器人手的观察者位置、肩关节、肘关节位置变换旋钮，以及机器人爪子的分合选项，左右手控制选项，镜头距离拉近，退出选项。如图4所示。

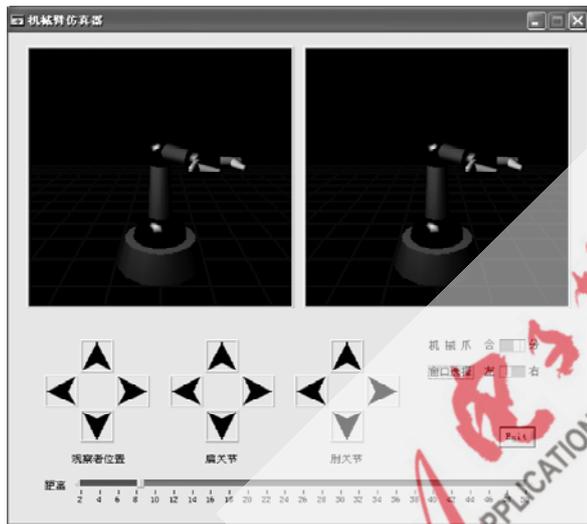


图4 控制机器人可视化仿真界面

3.4.2 实现机器人的动画

为了实现平滑的动画效果，OpenGL采用双缓存技术，双缓存是指位平面被分为前台缓存（显示缓存）和后台缓存（内存缓存）。后台缓存计算场景、生成动画，前台缓存显示后台缓存已经画好的画面。所以只有前台缓存才被显示。

因此当完整的画面在后台缓存中画好以后，就调用

SwapBuffers()函数，使其成为为前台可见。这样循环往复，隐藏了画图的过程，视频图像可以在人眼觉察不到的时间间隔交替出现，于是看起来所有的画面都是连续的。所以通过仿真界面可以控制机器人手的运动轨迹。

利用CVI和OpenGL的接口以及OpenGL在三维图形中的卓越表现，建立了机器人手的模型，设计了基于LabWindows/CVI和OpenGL的六自由度机器人手三维仿真系统，能够直观、正确、快捷地模拟现实机器人手的运动情况。作为进一步的研究，在此基础上，可以进行机器人视觉的仿真研究。

参考文献

- [1] 孙晓云. 基于LabWindows/ CVI的虚拟仪器设计与应用[M]. 北京:电子工业出版社,2005.
- [2] SHREINER D. OpenGL 编程指南[M].徐波, 译. 北京:机械工业出版社,2006.
- [3] 康凤举.现代仿真技术与应用[M].北京:国防工业出版社,2001.
- [4] 伍文伟.一种基于LabWindows/ CVI和OpenGL的六自由度机械手三维建模的方法[J].电子测量技术,2007, 30(8):75-77.
- [5] 李志国.基于OpenGL的光照处理技术及实现[J].计算机仿真, 2008,(3):221-222.
- [6] DONALD H, BAKER M. 计算机图形学[M].蔡士杰, 译. 北京:电子工业出版社, 2005.