

# 基于 Visual C++ 的水下机器人手爪作业过程仿真与实现

聂余满 魏铭旭 高理富 葛运建

(中国科学院合肥智能机械研究所机器人传感器实验室 安徽 合肥 230031)

**摘要** 针对水下机器人手爪作业过程的仿真实现,提出了一种界面友好的虚拟动态展示方法。该方法在已有水下机器人手爪的机械模型基础上,通过环境设置和相对位置调节,模拟水下作业过程;为改善虚拟作业过程的交互性并降低设计难度,在 Visual C++ 环境下用帧动画方法实现了动态过程的演示。该方法的思路和程序结构适用于类似应用场合。

**关键词** Visual C++ 水下机器人 作业演示 软件设计

## SIMULATION AND IMPLEMENTATION OF MANIPULATION PROCESS FOR AN UNDERWATER ROBOT HAND BASED ON VISUAL C++

Nie Yuman Wei Mingxu Gao Lifu Ge Yunjian

(Intelligent Robot Sensors LAB, Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei Anhui 230031, China)

**Abstract** This paper presents a method of developing a virtual animated demo with user-friendly interface for underwater robot hand manipulation process. Based on the made underwater robot hand model in SolidWorks, simulation of dynamic manipulation process was made by setting the circumstance and adjusting the relative position. The frame-by-frame animation method, which apparently gives the software development a better advantage of efficiency, is utilized with Visual C++ for the replay of the dynamic stimulation and a user-friendly interface. The principle of the method and the structure of the program has a general application to other similar occasions.

**Keywords** Visual C++ Underwater robot Manipulation demo Software design

### 1 引言

在水下机器人多传感器手爪感知系统的研究中,手爪作业状态对传感系统配置和信息感知具有重要影响<sup>[1]</sup>。为了根据手爪的作业特性来优化配置,需要展示手爪机构的几何外形、传动连接关系和其它物理特性。由于整个作业系统的复杂性,在系统完成调试前进行实际实验很不方便,因而考虑对作业过程进行仿真操作。

常规的三维仿真建模和虚拟显示需要对每一个活动对象分别进行设计,然后对这些对象进行实时变换,形成连续的动画过程。对于结构和运动关系都比较复杂的对象,模型建立和坐标变换难度通常较大。如目前在高性能图形和交互式视景处理方面常用的 OpenGL (Open Graphic Library), 只提供了简单形体的建模函数,而不提供用于描述复杂三维物体模型诸如汽车、人体、飞行器、分子之类的复杂三维物体模型的高层命令或者函数,用户必须从点、线、面等几何原型中构建自己的模型。这需要理解模型的坐标变换所对应的数学意义并自行编写变换函数,在实现多部件多关节复杂运动仿真时工作量大且实现困难。由于该水下手爪具有三个两关节连杆手指,关联运动部件多达 13 个,直接建立模型并进行仿真存在类似的问题。考虑到作业过程可以在机械设计软件中单步实现,综合使用机械设计软件 SolidWorks<sup>[2]</sup> 中虚拟操作与 Visual C++ 中帧动画相结合的方法,较好地解决了这个问题。基于设计模型的操作还避免了重新建模引起的误差,反映手爪作业状态精确直观。使用 Visual

C++ 编程的方法具有良好的通用性和可扩充性,为类似场合应用提供了参考。

### 2 水下手爪虚拟操作

水下机器人手爪的机械模型是在 Solidworks 中建立的。Solidworks 是 Windows 环境下集设计与运动轨迹校核于一体的机械 CAD 软件,采用非全约束的特征建模技术,具有建模速度快,直观,并能充分显示部件中相互之间的协调关系等特点。Solidwork 中也具备简单的动画工具,具有爆炸、反转、镜向和反转等功能,但是不能满足作业过程演示的需要,也不具有交互性。虚拟作业过程需要通过环境设置和位置调节来完成,如图 1 所示。

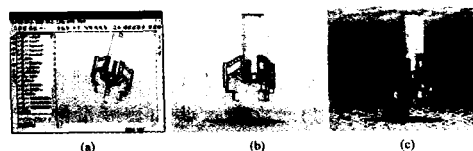


图 1 Solidworks 中环境设置

图 1 中, a 为基本的设计界面,其中的机械手三维模型可以进行开合与平移等操作。根据作业演示的需要,添加了操作对象(b 中的蓝色圆柱体)和显示相对位置的垫子(圆柱体下方灰

色片状物);为了表示水下作业环境,c 还使用了一幅海底世界的图片作为背景。

环境设置完成后,需要调节手爪本体、物体和垫子的位置进行虚拟作业。基本的操作方法如下:

(1) 移动手爪到物体正上方,调节垂直方向相对位置以保证物体位于可抓取范围;

(2) 选中手爪连接支撑杆(中心轴),下拉带动手爪往下做闭合运动趋近物体;

(3) 手爪指面与物体接触后,设置两者配合为表面相切,即保持夹持状态;

(4) 下移垫子,使之与手爪和物体的相对位置增大,然后调节垫子使之相对于背景的位置不变(意味着夹持物体的手爪在提升物体);

(5) 用类似的方法可完成移动和释放物体的操作。

从上述操作步骤可以发现,为了演示作业过程需要不断地调整相对位置,而且由于部分动作中相对位置需要反复改变,烦琐不直观且交互性差。特别是演示过程无法脱离设计环境,不适用与水下机器人手爪感知系统设计研究需要。为此,考虑制作可独立执行的演示软件并满足交互性要求。

### 3 软件设计

动画是利用人类视觉残留作用,快速播放一系列静态图像,产生动态效果。在计算机上实现动画的方法主要有实时动画法和非实时动画法。实时动画法包括简单动画法、双缓存法、色彩循环法等,在生成复杂图形时速度较慢,不适于复杂的机械手爪图形的绘制。非实时动画法有帧动画方法,主要思想是事先将动画序列的各帧绘制好存放在内存或文件中,然后按顺序播放各帧页面,从而产生动画效果。

使用 VC++ 实现帧动画方法,还可以用按钮控制动作过程实现交互。同时在水下机器人手爪作业演示软件中,通过 CPicture 类实现了直接从资源中读取 JPG 图像,解决了帧动画方法中通常存在的存储量较大的问题,提高了执行效率。在 VC++ 编译环境下生成的可执行文件大小为 5.44M(压缩文件大小 1.50M,欢迎到 ftp://202.127.200.100/Demo/下载)。软件制作过程为以下步骤:

#### 3.1 软件规划

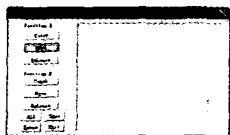


图 2 预设面板



图 3 软件界面

根据演示需要初步确定的控制界面如图 2 所示,最终的软件界面如图 3。界面上的按钮功能设定如下:

Catch: 指定位置抓取,位置可选 A/B。动作过程包括手爪从初始位置垂直下移趋近物体直至抓取范围→手腕位置保持固定,三指张开开始抓取→将物体提升到合适位置并保持垂直夹持状态;

Move: 保持垂直夹持状态移动物体到 B(或 A)位置;

Release: 手爪整体垂直下移,手指张开释放物体后手爪本体提升到合适位置;

All: 展示整个操作过程,包括从 A 到 B 及相反的过程;

Show: 倾斜一定角度,展示整个手爪结构;

Start/Pause: 在动作过程如需仔细观察,可以使用暂停功能;

Exit: 退出演示。

#### 3.2 准备帧

根据各个按钮的操作需求作初步的估计,确定大致需要 100~150 帧图片。取 110 张 JPG 图片,每张尺寸均为 733×587。按照动作过程和程序结构划分如表 1。

表 1 资源图片

序列	手爪动作	控制命令	调用帧序列	备注
001~024	←水平向左	Position A Catch	001~054	Bug1
025~035	↓垂直向下	Position A Move	055~076	
036~047	←←抓取物体	Position A Release	077~098	Bug2
048~054	↑垂直向上	Position B Catch	098~075	reverse
055~076	→水平向右	Position B Move	076~054	reverse
077~084	↓垂直向下	Position B Release	053~025	reverse
085~091	←→放置物体	All	001~098	
092~098	↑垂直向上	Show	099~110	Bug3
099~110	※旋转展示	Exit	none	

#### 3.3 程序结构

软件根据不同命令调用相应图片帧序列,显示动作过程,程序结构如图 4 所示:用户用各个按钮发出 Task Command,程序判断是哪一种命令获取相应的 Frame Serial,然后设定计时器,不断调用 On Timer 来增加/减少帧序号并显示对应的图像(Show Matching Frame)。

需要注意的是 VC++ 一般可以直接处理 bitmap 图像,但是 bitmap 图像文件会占用太多存储空间,在编译时效率很低,为此程序中使用了 CPicture 类<sup>[3]</sup>。CPicture 有一个 CComQIPtr < IPicture >, 指向 IPicture 接口,这个 Ptr 将通过调用 OleLoadPicture 做初始化。CPicture 提供了 IPicture<sup>[4]</sup>的所有方法,例如允许直接打开图形文件和从资源中打开。编程时把需要显示的 JPG 加入资源,并将资源格式定义为“IMAGE”,编辑资源文件为 IDR\_IMAGE1~IDR\_IMAGE110,然后使用函数 CPicture::Load( UINT nIDRes) 读取第 nIDRes 个资源对应的图片,使用函数 CPicture::Render( CDC \*, CRect, LPCRECT) 完成图片显示。

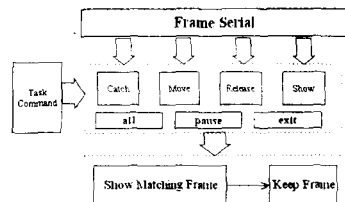


图 4 程序结构

整个显示过程的基本方法如下:

- 1) 初始设定 8 个时间事件的帧序列号起始值和终止值;
- 2) 等待用户发出按钮命令;
- 3) 获取按钮命令、判断命令,获取相应的帧序列号的起始值 Start 和终止值 End,令 Sequence = Start,若帧序号增加则 bIncrease = true,然后设定计时器;
- 4) 计时器处理:
  - i) 显示第 Sequence 帧图片;
  - ii) if bIncrease 为增则 Sequence ++,若 Sequence > End,则停止计时器,返回 2);
  - iii) else Sequence --,若 Sequence < End,则停止计时器,返回 2);
- 5) Pause 功能主要就是设定/停止计时器。(下转第 17 页)

定。输入层根据实际问题所确定的影响因素来定,输出层一般为1个,作为选址方案优劣程度的评价。

(4) 训练神经网络 根据已有方案请专家做出评价,得到评价值,即网络的期望输出。利用教师值对网络进行训练,确定网络的初始值和学习因子。学习因子在0.1~0.8之间。

a) 将第一步得到的配送中心选址决策的预选方案的指标数据作为输入,输入到训练过的网络中进行处理,得到结果。

b) 根据处理结果对评价结果进行排序,作为决策的依据。

## 4 应用实例

将指标隶属度输入到网络中,将专家值作为网络的期望输出,网络的初始权值取0~1的随机值,学习因子为0.4,  $Eps = 10^{-4}$ 。表2为经过量化和模糊处理,并求出隶属度的实验训练数据。经过多次迭代训练后得到的权值矩阵认为是最优矩阵,此网络可直接用于此类配送中心选址的评价选择。以训练结束后得到的最优矩阵为初始矩阵,将量化后的指标数据输入到经过训练的网络中,一次迭代后网络的输出为对方案对目标函数的符合程度,即对方案的评价值。表3中为经过初步筛选后的实验数据及其方案的评价结果及排序。

表2 评价样本

方案序号	地质条件	客户分布	候选地地价	供应商分布	运输距离	通讯条件	候选地面积	道路设施	专家评价结果
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	0.80	0.87	0.89	0.82	0.78	0.80	0.75	0.33	0.79
3	0.67	0.93	0.22	0.75	1.00	0.80	0.49	0.66	0.74
4	0.92	0.80	0.89	0.92	0.89	0.80	1.00	1.00	0.81
5	0.87	0.93	1.00	1.00	1.00	1.00	1.00	1.00	0.96
6	0.80	0.72	0.89	0.82	0.89	0.80	0.75	1.00	0.83
7	0.67	0.72	0.67	0.66	0.67	0.60	0.49	0.66	0.69
8	0.72	0.80	0.78	0.75	0.78	0.80	0.75	0.66	0.75
9	0.60	0.60	0.56	0.58	0.56	0.60	0.49	0.66	0.58
10	0.47	0.47	0.44	0.41	0.44	0.40	0.49	0.35	0.51

表3 评价方案和结果

方案序号	地质条件	客户分布	候选地地价	供应商分布	运输距离	通讯条件	候选地面积	道路设施	评价结果	排序
11	0.40	0.40	0.33	0.33	0.33	0.40	0.49	0.35	0.34	5
12	0.08	0.93	0.56	0.92	0.89	0.60	0.24	0.33	0.69	1
13	0.67	0.60	0.89	0.82	1.00	0.80	0.75	0.29	0.48	3
14	0.32	0.40	0.67	0.33	0.33	0.80	0.75	0.35	0.44	4
15	0.87	0.72	0.89	0.92	0.89	0.40	0.49	0.29	0.54	2

从以上计算结果可以看出,方案12最优,方案11最差,这也与各方案的指标因素特点是基本一致的,根据此结果可以进行配送中心的选址的决策。

## 5 结束语

配送中心选址在实际中是非常复杂的问题,各因素间相互

影响,模糊神经网络方法为解决此类问题与其它方法相比有以下优点:模糊神经网络是处理此类非线性关系问题的很好的工具,因此计算过程相对简洁;模糊神经网络的输入是通过模糊方法将指标量化并计算出隶属度,而网络的训练则是自适应的过程,评价结果较为客观。此种方法的关键即是定性因素的量化和获得大量的历史数据来对网络进行训练,如果可以保证以上两点,那么此种方法的优越性就会得到更大的体现。

## 参 考 文 献

- [1] 魏光兴,“物流配送中心选址方法综述”,《物流与交通》2005/09.
- [2] 徐杰、田源、汝宜红,“物流中心选址的影响因素分析及案例”,《物流技术》,2002,2.
- [3] 丁浩、李电生,“城市物流配送中心选址方法的研究”,《华中科技大学学报(城市科学版)》,2004,3.
- [4] 陈非、敬忠良、姚晓东,“一种模糊神经网络的快速参数学习算法”,《控制理论与应用》,2002,8.
- [5] Kong S G, Kosko B. Adaptive fuzzy system for backing up a track and - trailer. IEEE Trans. Neural Networks, 1992, 3 (2): 211 ~ 233.
- [6] 邵举平、董绍华,“综合型物流中心功能设计和选址模型”,《物流技术》,2004,12.
- [7] 诸静,模糊控制原理与应用,北京:机械工业出版社,1995,46 ~ 58.
- [8] 吴坚、史忠科,“基于遗传算法的配送中心选址问题”,《华南理工大学学报》,2004,06.

(上接第9页)

### 3.4 需要注意的问题

(1) 为了保证获取图片尺寸质量的一致,应使用专用图形捕捉软件,如 SnagIt7.1.1<sup>[5]</sup>;

(2) 在处理图片时要去除一些冗余或不足。在3.2步骤中得到的110张图片,其中就有3张含缺陷(表1中备注栏已标注Bug),为不影响显示效果使用前后图片复制替换,实际显示时基本没有差别;

(3) 为编程中调用方便,在VC++资源中应尽量使图片资源序列号与原文件序列按次序对应,可能需要手动或者自己写程序进行批量修改。

## 4 结 论

演示软件直观生动地展示了水下机器人手爪的作业过程,较好地满足了交互性要求。综合使用SolidWorks中虚拟操作和Visual C++帧动画的方法,有效降低了软件开发难度,提高了制作效率。使用Visual C++制作软件还具有灵活性、可扩展性和可维护性,为复杂机构的操作演示、产品展示和作业规划等应用场合提供了参考。

## 参 考 文 献

- [1] 许德章,水下机器人手爪信息感知系统的研究[D],博士学位论文,合肥:中国科学技术大学,2005:101~107.
- [2] <http://www.solidworks.com>.
- [3] <http://www.csdn.com.cn/program/6000.htm>.
- [4] <http://www.csdn.com.cn/program/5559.htm>.
- [5] <http://www.techsmith.com>.