第33卷 第4期 计 算 机 工 程 2007年2月
Vol.33 No.4 Computer Engineering February 2007
·工程应用技术与实现· 文章编号：1000—3428(2007)04—0241—04 文献标识码：A 中图分类号：TP23

三维雕刻机的远程控制技术研究及系统实现

董 刚，陈一民，邹一波
(上海大学计算机工程与科学学院，上海 200072)

摘 要：介绍了雕刻机三维模型设计制作中的主要技术，并对虚拟场景的构建和优化作了说明。在三维雕刻机的模拟仿真中，对工件结构设计和NC代码解释执行等技术作了深入探讨。通过WINSOCK技术实现了对雕刻机的远程控制，并采用自适应端到端的QoS控制技术，在三维雕刻机的远程控制中实现了流畅的视频传输。

关键词：远程控制；雕刻机；Open Inventor

Research and Realization of Remote Control of 3D Engraving Plotter

DONG Gang, CHEN Yimin, ZOU Yibo
(School of Computer Engineering and Science, Shanghai University, Shanghai 200072)

【Abstract】 This paper introduces the main techniques about the design and development of the model of the 3D engraving plotter. The methods of machine geometry modeling and scene optimizing based on the Open Inventor are proposed. It also describes the design of the structure of working object and the interpretation of NC codes for the 3D engraving plotter. It accomplishes the remote control by the socket network programming and implements the smooth video transfer by using the adaptive end-to-end congestion control in this system.

【Key words】 Remote control; Engraving plotter; Open Inventor

网络技术的发展，尤其是Internet的广泛应用，为远程监控的研究与应用提供了基础。远程控制中的被控对象除具有一些基本特点外，还具有与外界进行良好信息交互的特点，除了有本地控制能力外，还有远程控制能力。1994年，美国南加州大学Ken Goldberg等人建立的Mercury Project机器人系统，标志着基于Internet远程控制机器人的诞生。此后，澳大利亚Wollogong大学的Roboty，德国的Net-Robot，Rovetta研制了一种远程医疗机器人系统。在国内，清华大学开发了基于事件的网络机器人遥操作系统，它采用基于事件和图形预测仿真的直接控制方法，实现了Internet上的多用户双臂遥操作系统。上海交通大学也开发了基于网络的机器人遥操作系统，被控机器人是一台Adept 604-S型工业机械手。最近中科院“863”项目开发出了消防机器人。但总体说来国内在这方面的研究起步较晚，与欧美日还有不小的差距。本文所设计的系统，通过Internet实现对远程雕刻机的控制。为保证被控设备的安全、可靠工作，加入了远程视频监视及虚拟雕刻机场景仿真。

1 远程控制系统的结构

远程控制系统的结构如图1所示，由远端服务器、视频处理器、雕刻机控制器、摄像机、雕刻机本体和客户端(包括仿真模块、视频处理监视模块、远程控制模块)组成。

用户在客户端发送控制命令，通过Internet传给远端的服务器(其中雕刻机控制器也可包括在服务器中)，再传给远端雕刻机，雕刻机控制器对控制命令加以解释执行，从而控制雕刻机完成相应的动作。另一方面，由远端的摄像机把现场的图像采集起来，通过视频处理器压缩处理数据，通过服务器、Internet传回给客户端的用户，为达到较流畅播放视频的效果，在发送端和接收端之间采用了端到端的自适应QoS控制。考虑到视频传输所带来的延时的不确定性，为正确了解远端雕刻机的工况，在本地建立了虚拟的雕刻机环境，虚拟雕刻机只需通过回传的反馈信号(NC代码)就可模拟真实雕刻机的加工过程。因此只要保证回传的反馈信号准时到达控制端，就可实时了解雕刻机的工况。

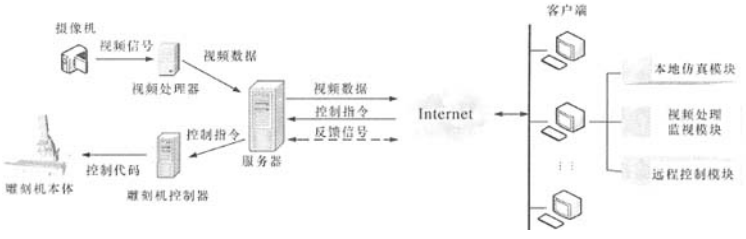

图1 远程控制系统结构

2 虚拟雕刻机场景构建

在构建虚拟雕刻机场景时，选用的是TGS公司开发的Open Inventor三维开发包，它由完整的用C++编写的大型类库组成，支持面向对象的设计方法，可以实现对象的造型、属性描述、动画表现等一系列功能。我们开发了基于Open Inventor的虚拟雕刻机。

虚拟雕刻机由众多的零部件组成，为了从不同角度及深度展示雕刻机的结构及组成，需要按一定的规则把各种零部件组合起来。为此根据三维场景数据规则，构造出虚拟雕刻

作者简介：董 刚(1978－)，男，硕士生，主研方向：机器人控制，多媒体技术；陈一民，教授、博导；邹一波，硕士生

收稿日期：2006-03-13 **E-mail**：gdong@mail.shu.edu.cn

机的场景图，如图 2 所示。

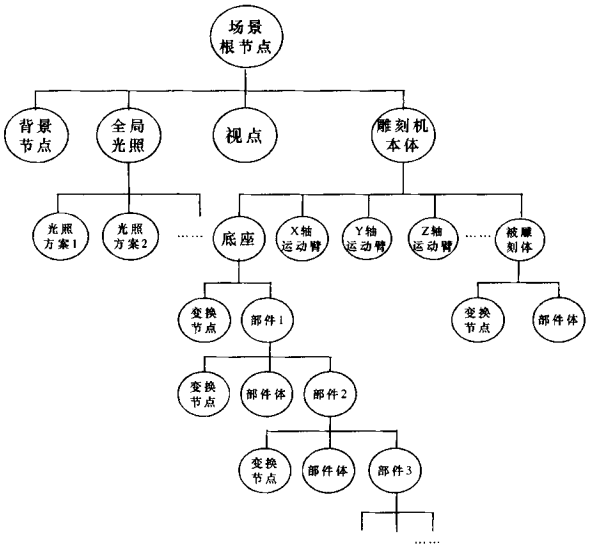

图 2 虚拟雕刻机场景

在场景图的最上层，是虚拟场景的根节点，所有的雕刻机仿真图形场景都是从这个节点中衍生出来的。场景图第 2 层的最左端是背景场景节点，如周围的环境特性。其次是全局光照节点，定义了对整个场景的光照方案。接下来是视点节点，它定义了雕刻机的全局视点，通过对视点节点属性的设置，用户可以从不同视角对雕刻机的工况进行观察。然后，就是雕刻机本体节点。由于雕刻机是由各部件构成且各部件间存在联动关系，因此利用分组节点构成层次结构的虚拟系统：雕刻机本体由变换节点、部件、下层部件……直到最底层的部件体层层嵌套而成。每个部件、下层部件都是一个分组节点。部件体是虚拟雕刻机最基础的组成部分，在部件组节点中包括部件体的几何位姿变换节点和零件的几何形体。

3 虚拟雕刻机的几何建模

我们用 UniGraphics 软件建立虚拟物体的几何模型，再将它转换成 VRML97 格式。VRML 作为一种景象(scene)描述语言，提供了 6+1 度的移动空间，可以沿着 3 个方向移动，也可以沿着 3 个方位旋转。根据实测的雕刻机各部件的具体数据进行虚拟设计，包括零部件的三维建模和虚拟装配。在设计过程中，采用基于特征的参数化造型方法，创建的模型部件主要包括：底座，X 轴运动臂，Y 轴运动臂，Z 轴运动臂和其它的一些辅助部件。

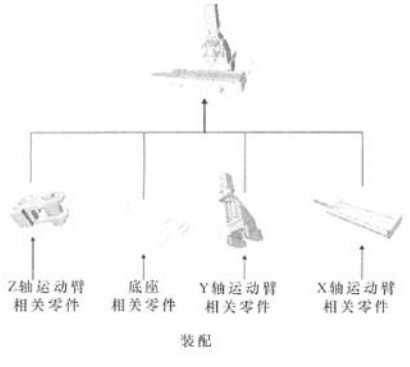

图 3 雕刻机装配过程

对雕刻机的建模实际上可分为运动学建模和几何建模两部分。但是几何建模与运动学建模实际上是密不可分的，必须根据具体情况通盘考虑。例如为了使雕刻机动起来，必须对雕刻机的每个动作部件单独建模，生成几何模型。但同时还必须使每个几何模型保持其空间位置信息(亦称为装配)，这样才能使各个连杆有机地组合在一起。整个装配过程如图 3 所示。

4 三维雕刻机的模拟仿真

4.1 工件的几何模型

雕刻机的钻头在 x 轴、y 轴、z 轴发生移动，被加工的毛坯可视为立方体。毛坯的上表面表示成 m×n 的矩阵网格，如图 4 所示。

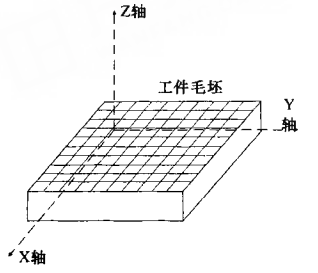

图 4 工件毛坯的矩阵网格

每个交叉点称为节点，其 x 坐标和 y 坐标是固定不变的，改变的是它的深度(z 坐标)。在仿真过程中，通过不断地改变节点的深度来表示毛坯被加工的实际情况。系统在雕刻机运动过程中不断返回雕刻机钻头及被刻物体的实时位置，从而可以获得钻头目前所对应的矩阵网格中的节点。然后修改对应节点的深度得到当前点的被刻后的效果。

雕刻机钻头的运动轨迹是由直线段组成的，每一段由钻头的起点和终点来描述。计算机计算一次钻头在此时间中的运动轨迹，修改钻头所经过的节点的深度，并重新构成当前毛坯的形状，显示结果。仿真过程实际上就是一个计算机不断计算、修改节点深度的过程。

4.2 NC 码的解释执行

由于用户输入的是以文本文件形式保存的标准数控加工指令 NC 代码，因此系统在加工时，需要把加工文件中的加工指令进行解释以生成加工数据链表，数据链表中的每个节点对应一条 NC 代码指令，节点内部记录的数据包括：

指令类型，标准 NC 代码指令有多个大类，目前实现了主要的 G 指令和 M 指令，如直线进给 G01，圆弧进给 G02、G03，停止指令 M30，开始加工指令 M05 等。解释器要识别出指令关键字 G、M，再识别后面所带的数字，从而确定调用的是哪一条指令。

指令参数，不同类型的指令有不同的参数列表，确定当前 G 指令的类型之后，就要识别出与它对应的指令参数关键字，包括 XOZ 平面坐标关键字 X、Z；I、J 等，然后再读取关键字后的参数值。

此外，G 代码文件每一行中“;”之后的内容都是注释，解释器必须正确识别出注释部分以将其忽略。

代码解释子模块的流程如图 5 所示。

用解释器将指令解释为特定的格式，存入已定义的 mNCContent 数组中，将数组中的指令一条一条地同时在虚拟雕刻机和现实雕刻机中执行。数组 mNCContent 中每个数据单元的格式如下：

```
typedef struct
 {
 int mLineNum;      //NC 代码行号
 char mOperator;    //操作码
 int mOperatorNum; //操作号
 float mX;          //X 坐标
 float mY;          //Y 坐标
 float mZ;          //Z 坐标
 float mI;          //圆弧中心 X 坐标
 float mJ;          //圆弧中心 Y 坐标
 float mF;          //进给量
}
NCCODE;
```

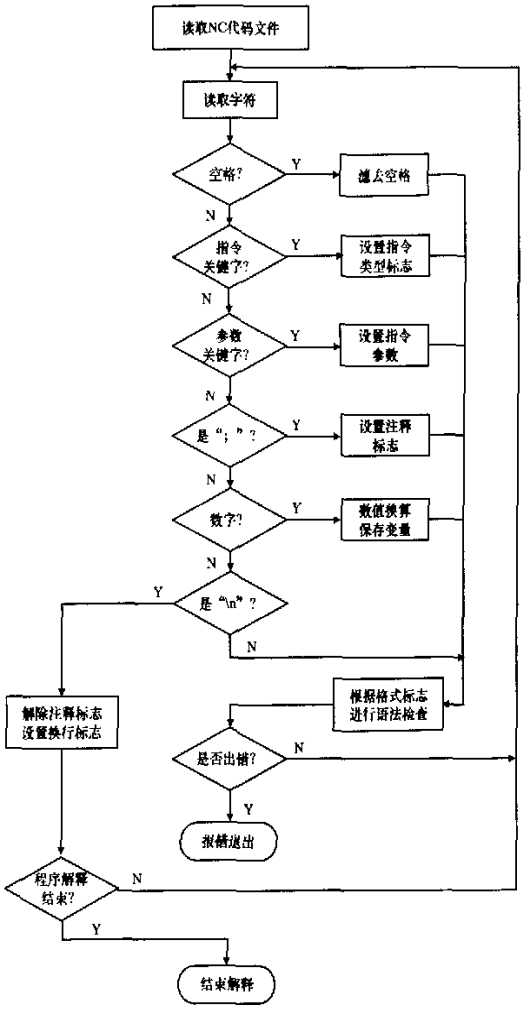

图 5 代码解释子模块流程

5 三维雕刻机的远程控制

5.1 三维雕刻机的控制

通过雕刻机控制器提供的 Windows 运动函数动态链接库，就可以实现运动控制器的各种功能。在控制过程中调用了许多函数，如：GT_Axis，设置指定轴为当前轴；GT_AxisOn，使当前轴处于打开控制通道状态，并使轴驱动器驱动允许；GT_GetAtPos，获得当前轴的实际位置值；GT_LmtSns，设置限位开关有效电平；GT_SetSynAcc，设置坐标系运动中轨迹段的合成加速度等。

5.2 远程控制

我们通过封装 MFC 下的 CAsyncSocket 类来实现 Client/Server 模式的网络远程控制。CAsyncSocket 类在较低的层次上逐个封装了 WinSocket API，保持了原有的异步特性；而 CSocket 类是由 CAsyncSocket 继承而来的。CAsyncSocket 类数据通信的实现步骤如下(通常是客户机/服务器结构)：

服务器：Construct → Create( ) → Listen( ) → Accept( ) → Receive( ) → Close( )

客户端：Construct → Create( ) → Connect( ) → Send( ) → Close( )

三维雕刻机的客户端和服务器端分别抽象为两个类。

客户端新建一个类 CClientSock。这个类继承了 CAsyncSocket 类，并重载了 OnSend( )，OnConnect( )，OnClose( )事件。其中，OnSend( )是当 socket 发送数据时触发，OnConnect( )是当 socket 连接时触发，OnClose( )是当 socket 关闭时触发。

在服务器端，新建两个类 CServerSocket 和 CNewSocket。其中，CServerSocket 是服务器端 Socket，用来处理来自客户端的连接，而 CNewSocket 是用户连接之后服务器用它的实例去处理与客户的消息交换。

CServerSocket 类重载了 CAsyncSocket 的两个函数 OnAccept( ) 和 OnClose( )，CNewSocket 类重载了 CAsyncSocket 的 OnReceive( )函数。

在开始进行连接时，CServerSocket 设置感兴趣的事件为“连接”，有客户连接之后，在 OnAccept( )的处理中建立 CNewSocket 对象。

通过网络不仅可以传送单条指令，而且还能传送整个 NC 代码文件。方便用户采用不同的操作方式。

6 视频传输控制

6.1 模型结构

为了更好地方便用户进行远程的控制操作，给人以良好的“沉浸”感，有必要使用摄像头捕获三维雕刻机工作现场的视频图像，通过网络实时地将机器运动状态传输给远程的控制者，方便其实时控制。由带宽控制视频质量，当带宽充足时，提供较好的视频质量；当带宽不够时，通过适当降低视频的帧数以及减小视频图像的尺寸来满足视频传输实时性要求。

视频传输模型如图 6 所示。

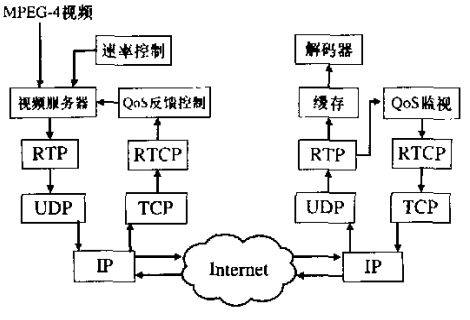

图 6 视频传输模型

6.2 控制算法

当网络流量波动比较大时，将会导致数据传输速率的剧烈变动。为了平滑这种波动，采用了一个低通滤波器。

当前要平滑的平均丢包率 λ 为

$$\lambda = (1-\alpha)\bar{\lambda} + \alpha b \quad (1)$$

其中，b 为新统计的丢包值，$\bar{\lambda}$ 为上一次计算得到的平均丢包，α 为平滑系数，且 $0 \le \alpha \le 1$。增大 α，将会增加新统计丢包值 b 在平均丢包率 λ 中的权重。

通过接收端获得的短期平均丢包率作为判断网络拥塞状况的度量，根据平均丢包率的情况来决定增加、保持还是减少网络带宽。其中设定了平均丢包率的上限和下限来作为判断的标准。

上限 λ_h 为接收端能够承受的丢包率的最大值，为了减小 QoS 的剧烈变动，下限 λ_l 设为较低的数值。根据试验结果来设置参数，这里设置 λ_h=0.05，λ_l=0.015。

然后根据下面的算法作出判断：

```
if decision = DECREASE then
    r = max{r × μ, min_bw}
else if decision = INCREASE then
    r = min{r + v, max_bw }
```

其中，r 为当前获得的传输速率，min_bw 和 max_bw 为该数据流所能获得的最小和最大带宽，μ 为乘性参量，v 为加性参量。为了根据当前网络状况，更加灵活地改变码流，将 μ 和 v 设为变量，以期达到快速平滑网络波动的效果。

$$v_i = \min\left(v_{i\_add}, v_{i\_exp}\right) \quad (2)$$

其中 v_{i_add} 和 v_{i_exp} 分别为采用两种不同方法所得到的递增步长。

$$\mu = 1 - \sqrt{l} \quad (3)$$

其中 l 为丢包率。

$$r_{TCP} = C\frac{M}{t_{RTT}\sqrt{l}} \quad (4)$$

其中，M 为数据包的大小，l 为丢包率，t_{RTT} 为数据往返的时延，C 为常数，一般取 1.22。由式(4)可方便地确定 l 的大小，进而得到 μ 的值。

当然，为了保证远程控制的稳定性，不管网络带宽如何变化，控制流传输的优先级始终高于视频流传输。

7 实现及运行结果

在具体实现过程中，将视频自适应传输算法封装成一个滤镜 Filter，然后加入到 DirectShow 程序中。DirectShow 程序是以滤镜为基本单位组装而成的，每个滤镜是对数据的不同处理。

系统在运行过程中，在网内拥有独立 IP 的计算机上，实现了较好的实时远程控制并获得了较流畅的视频传输。图 7 给出了真实雕刻机(右)和虚拟雕刻机(左)在工作中的比较。

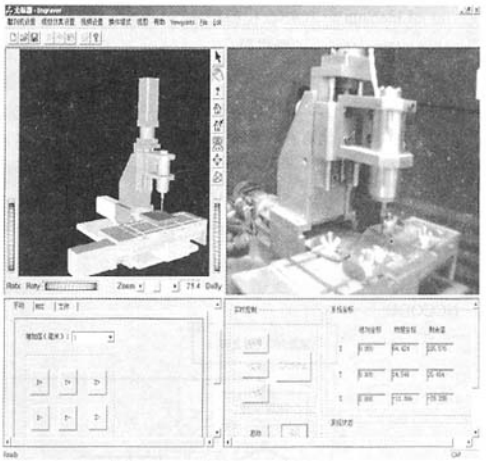

图 7 真实雕刻机与虚拟雕刻机比较

8 结论及展望

本文采用了 Winsock 技术，实现了雕刻机的控制指令及视频图像的远程传输。对控制指令(NC 码)引入了解释执行技术；采用端到端的自适应 QoS 技术，保证了监视视频播放的流畅性。通过综合运用几何建模，三维场景绘制，我们设计了一个基于 GT-400 的虚拟雕刻机系统，为提高其仿真运行速度采用了细节层次技术对其进行优化。并通过实例说明了系统的有效性。

参考文献

1 Sisalem D, Wolisz A. Towards TCP-friendly Adaptive Multimedia Applications Based on RTP[C]//Proceedings of the 4th IEEE Symposium on Computers and Communications. 1999-07: 166-172.

2 Busse I, Deffner B. Dynamic QoS Control of Multimedia Applications Based on RTP[J]. Computer Communications, 1996, 19(1): 49-58.

3 Wernecke J. The Inventor Mentor: Programming Object-oriented 3D Graphics with Open Inventor[M]. Addisson Wesley, 1998.

4 董振亚，张拥军，彭宇行. 基于 RTP 的 MPEG-4 视频传输[J]. 计算机应用研究，2003, 20(7): 52-55.

5 牛文博，刘 进. 一种数控铣床的仿真算法[J]. 计算机辅助设计与图形学学报，2002, 14(5): 464-467.

(上接第 240 页)

除线程切换过程中的开销，从而提高硬件多线程处理器的性能。还根据流水线的执行模型，准确地分析了线程切换频度、流水线深度对硬件多线程处理器性能的影响，同时进行了模拟实验验证，对下一代支持硬件多线程的微处理器设计有一定的参考价值。

在研究过程中，也发现线程数目的多少对处理器的性能也有一定的影响，硬件线程数目设置过少，会发生流水线暂停的情况，或多或少地对性能造成影响。因此，多线程处理器中线程数目的设置是一个值得研究的课题。

参考文献

1 IXP2400 Hardware Reference Manual[Z]. Intel Corporation, 2002.

2 Rudd K W. General Purpose VLIW Processors Using Replay Buffers[D]. Stanford, CA, USA: Stanford University, 1999.

3 Mulder J M. Tradeoffs in Processor-architecture and Data-buffer Design[D]. Stanford, CA, USA: Stanford University, 1988.

4 谭章熹，林 闯，任丰源，等. 网络处理器的分析与研究[J]. 软件学报，2003, 14(2).

5 吴 闻，李雪莹，许榕生，等. IXP2400 网络处理器及其微引擎中多线程实现的研究[J]. 计算机工程与应用，2004, 40(9).