

47-49

# 虚拟现实技术在微机上的实现

中国科学院合肥智能机械研究所(230031) 钱 玮 路 巍 李劲松 TP391.9

**摘 要:**虚拟现实技术在微机上的实现方法及虚拟现实系统的工作原理,并且从硬件和软件两方面说明了虚拟现实技术在微机上的实现的要点。

**关键词:**虚拟现实 PCI总线 微机

虚拟现实(VR)技术,也称临场感技术,它通过环境模拟使人产生身临其境的感觉。其中视觉临场感技术是非常重要的环节。一般来说,虚拟现实视觉技术是通过双眼图像系统使人在视觉感受上具有观察实际物体的三维空间立体感觉。目前虚拟现实视觉技术主要有双眼头盔显示方式和分时双眼显示方式。本文介绍了分时双眼显示系统的原理及其在微机上的实现方法。

## 1 分时双眼显示原理

左右摄像机的图像在显示屏上分时交替显示,观察者戴上1种液晶光阀眼镜进行观看,左右2眼的液晶光阀分时轮流进行打开和关闭。当液晶光阀的开关与屏幕上显示的左右2个摄像机的图像交替显示的频率同步时,即屏幕上显示左摄像机的图像时,左眼的液晶光阀打开,使观察者左眼可以看到左摄像机传到屏幕上的图像,而这时右眼液晶光阀关闭使观察者右眼看不到任何图像,反之亦然。这样观察者左眼就始终只能看到左摄像机传来的图像,右眼始终只能看到右摄像机传来的图像。当这种左右交替的速度大于60Hz时,由于人眼“视觉暂留现象”存在,就能使人左右2眼分别看到2组分别来自2台放置位置和视角与人眼相似的摄像机的连续流畅的图像,从而使人产生很好的立体视觉效果<sup>[1]</sup>。

## 2 分时双眼显示系统的硬件构成

本系统由数据采集(见图1)和显示(见图2)二部分组成。物理结构是基于PCI总线的2块微机插卡。数据采集部分是由同步控制器控制的2台摄像机实现的,同步拍摄供给左右2眼的电视图像,2路电视信号分别进行A/D转换(AD8708芯片)和数字解码(7196芯片),在同步切换控制器控制下,从左摄像机来的视频数据中取奇场数据(CCIR制式下1个视频帧由奇偶两场信号组成),从右摄像机来的视频数据中取偶场数据。由于2摄像机已同步,因此奇偶场数据虽然来自不同的视频流(左右2摄像机视频信号),但在信号的时域结构上来看可以合成1帧图像。这样来自不同视频流的场频数据在同步切换控制下重新合成为帧频数据。经PCI总线接口芯片打包

后,在计算机DMA控制下将数据实时送至ET4000显卡的帧存中。在显卡的控制下图像帧按奇、偶场顺序交替在显示屏上显示。观察者戴液晶光阀眼镜,在分时同步切换的控制下,使左眼液晶光阀只在屏幕上显示奇场图像时打开。右眼液晶光阀只在屏幕上显示偶场图像时打开。

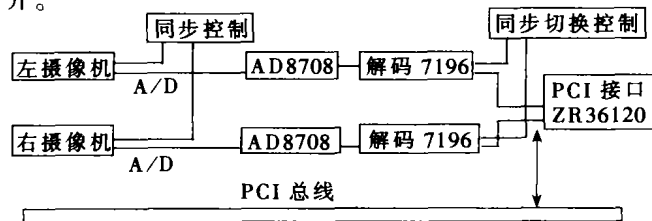


图1 数据采集部分

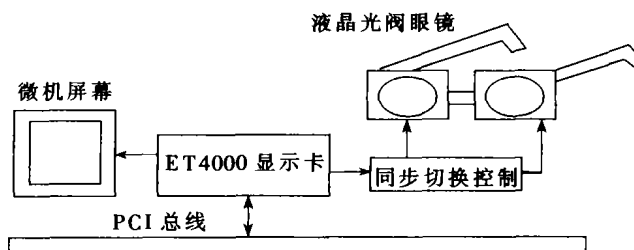


图2 显示部分

## 3 系统软件的编程

### 1. 硬件系统编程控制

硬件系统的控制编程采用Watcom C保护模式编程技术,这种选择基于以下几方面原因。

(1)PCI接口芯片ZR36120的端口控制采用内存映射方式,映射地址是32位真实物理地址(与实模式16段址+16位偏移位不同)。

(2)为了达到系统的设计要求,经ZR36120接口进入主机的图像数据要直接进入显示缓存,而在实模式下,显示缓存的写入是通过一种页映射方式完成的,需置段选择寄存器,速度无法达到要求。

(3)Watcom C编译系统是由加拿大WATCOM系统软件公司以新的美国国家标准ANSI X3.159为基础于1992

## 应用奇葩

年推出的具有 32 位编程功能的软件包,它不仅能访问机器中的所有内存,而且程序的执行速度亦能成倍提高,有利于提高系统的性能。

### 2. 系统控制软件编程要点

(1) 系统控制软件流程图(如图 3 所示)

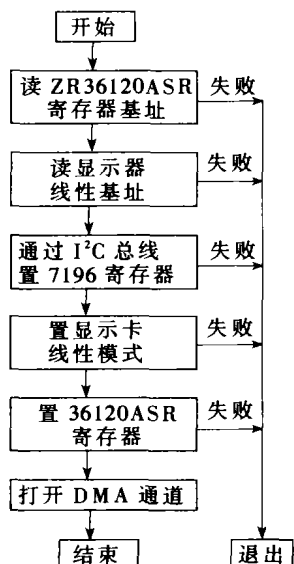


图 3 系统控制软件流程图

(2) ZR36120 的 ASR 寄存器编程说明

ZR36120 的 I/O 口是通过内存映射方式工作的,因此,须访问 PCI 的配置空间,获取 ZR36120 的 ASR 寄存器基址,利用该基址加偏移地址生成 32 位的物理指针就可以对其寄存器进行读写操作。

ZR36120 共有 20 个 32 位的寄存器,统称为 ASR<sup>[2]</sup>,通过这组寄存器可以实现对整个采集卡的硬件控制,其中第 18 个寄存器为 I<sup>2</sup>C—BUS 寄存器,通过该寄存器的读写,可以控制连接在 I<sup>2</sup>C 总线上的设备,如解码芯片 7196 以及同步控制器。

(3) 解码芯片 7196 的控制说明

7196 共有 64 个寄存器,分为二部分:

前 32 个为编码控制,后 32 个为 scaler 控制。

编码控制包括:视频制式选择(PAL、NTSC)、同步信号输出方式、数字视频格式(YUV、RGB)、象素格式(64K、16M)。

scaler 控制包括:图像尺寸、采样精度等。

7196 寄存器的读写格式如图 4<sup>[2]</sup>:

S	slave address	A	Subaddress	A	DATAO	A	DATAN	A	P
---	---------------	---	------------	---	-------	---	-------	---	---

图 4 7196 寄存器读写格式

S = start condition

slave address = 0100000 或 010100001

A = acknowledge generated by the slave

subaddress = subaddress byte (0 ~ 64)

DATA = data byte

P = stop condition

X = 0 write

X = 1 read

(4) 显示卡线性模式编程

PCI 总线的显示卡有二种显示方式:①基于段选择器的显示方式;②连续线性地址显示方式。我们采用的是第 2 种方式。

由于 ET4000 显示卡目前比较多,并且有大量的文章对其进行介绍和分析,因此本文对 ET4000 卡的编程控制介绍从略。

(5) 硬件控制关键部分程序清单

①访问 PCI 的配置空间,获取 ZR36120 的 ASR 寄存器基址的子程序,通过 PCI 中断 B1 读取 ASR 基址,配置空间说明见文献[2]。

```

long get_ASR_Base()
{
    union REGS r;
    unsigned long ab;
    r.h.ah = 0xb1;
    r.h.al = 0x0a;
    r.w.di = 0x10;
    int 386(0x1A, &r, &r);
    return(r.x.ecx);
}
  
```

② ASR 寄存器的读写函数

```

Write_ASR(int index, long value)
/* index 是 ASR 寄存器索引, value 是写入的值 */
{
    long * ASR;
    ASR = get_ASR_Base();
    *(ASR + index) = value;
}

Read_ASR(int index) /* index 是 ASR 寄存器索引 */
{
    long * ASR, t;
    ASR = get_ASR_Base();
    t = *(ASR + index);
    return(t);
}
  
```

③ 7196 寄存器的读写函数

7196 寄存器的读写通过 I<sup>2</sup>C 总线来完成。I<sup>2</sup>C 总线是通过 ZR36120 的 18 号寄存器进行操作。具体操作函数如下:

```

long * ASR_base; /* ASR 基址,全局变量 */
IIC_start(void) /* 设置起始条件 */
{
}
  
```

```

* (ASR_base + 0x44) = 3;
delay(1);
* (ASR_base + 0x44) = 1;
delay(1);
return;
}
IIC_stop(void) /* 设置结束条件 */
{
* (ASR_base + 0x44) = 1;
delay(1);
* (ASR_base + 0x44) = 3;
delay(1);
return;
}
IIC_bit_write(unsigned char data_bit) /* 写 7196 寄存器
*/
{
if(data_bit == 0)
{
* (ASR_base + 0x44) = 0;delay(1);
* (ASR_base + 0x44) = 1;delay(1);
* (ASR_base + 0x44) = 0;delay(1);
}
if(data_bit == 1)
{
* (ASR_base + 0x44) = 2;delay(1);
* (ASR_base + 0x44) = 3;delay(1);
* (ASR_base + 0x44) = 2;delay(1);
}
}

```

```

}
IIC_bit_read(void) /* 读 7196 寄存器 */
{
unsigned char read_value;
* (ASR_base + 0x44) = 2;delay(1);
* (ASR_base + 0x44) = 3;delay(1);
read_value = * (ASR_base + 0x44);
if(read_value >= 2) read_value = 1;
else read_value = 0;
* (ASR_base + 0x44) = 2;delay(1);
return((int)read_value);
}

```

#### 4 结束语

本系统设计中涉及的软硬件内容十分广泛,由于篇幅限制,不能每一部分都作深入讨论,仅能对所设计的虚拟现实视觉系统的工作原理和实现思路做简单描述。这些描述虽然简单,但力求准确,读者通过本文可以对分时双眼虚拟现实视觉系统的实现和组成有一个比较全面的了解。

#### 参考文献

- 1 Steve B. Virtual Reality in Scientific Visualization. IEEE Computer&graphics,1993;17(6)
- 2 ZORAN.ZR36120 Data Sheet,1995
- 3 Philips. DATA HandBook,IC22.1995
- 4 ZORAN.ZR36060 Data Sheet,1996

(收稿日期:1998-08-10)

(上接第 44 页)

#### 3.3 并行系统设计

并行管理系统主要是完成对并行工作站的管理,它大致可分为接收数据、并行查询、统计预测和数据维护等四个功能,它相对于用户数据联机分析系统来说是后端。并行系统设计如图 6。

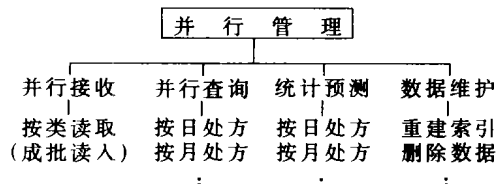


图 6 并行系统结构设计

#### 4 结束语

数据仓库应用系统的开发是一项非常复杂的技术工作,本文为建造中小型的数据仓库系统提供了一套模式。我们已用本文的设计思想在计算机网络上进行了学生成绩数据仓库管理系统的实验,为建造符合实际需要的数

据仓库开辟了一条有效途径。

#### 参考文献

- 1 黄璇. 数据库技术的发展方向. 计算机工程与应用. 1995, 31(5):1
- 2 杨利,周兴铭,郑若忠. 并行数据库系统的体系结构. 计算机科学,1994;21(4):42
- 3 杨利,周兴铭,吴涛. 并行查询中的进程分配与调度. 计算机科学,1995;22(6):26
- 4 郭宜斌. 数据仓库技术的基本概念和发展现状. PC 世界, 1996;(4):26
- 5 于戈,张斌. 数据仓库管理中的若干关键技术. 计算机科学,1997;(2):31
- 6 唐韶华. 基于 C/S 结构的网络分布式数据库应用. 微型计算机,1996;16(2):13
- 7 王晓蔚,孙志晖. 基于客户/服务器的 CMIS 分布式数据库系统设计. 微型计算机,1996;16(5):13

(收稿日期:1998-09-10)