

doi: 10.3969/j.issn.1006-1576.2010.03.031

DirectDraw 下 MMX 指令视频显示调节实现

宋方伟^{1,2}, 刘勇¹, 聂诗良², 王能¹

(1. 中国兵器工业第五八研究所 军电部, 四川 绵阳 621000; 2. 西南科技大学 信息工程学院, 四川 绵阳 621010)

摘要: 针对 DirectDraw 视频调节显示接口函数存在的不足, 采用多媒体增强指令集 (MMX) 实现 DirectDraw 视频调节接口。从 DirectDraw 的一般工作流程、双缓冲显示原理和 DirectDraw 显示实现对 DirectDraw 视频播放器进行原理分析设计; 介绍了 MMX 技术的使用流程; 并进行了试验分析。试验表明, 该方法运行速度快、效率高、视频流畅。

关键词: DirectDraw; MMX; 视频调节

中图分类号: TP302.7 **文献标识码:** A

Implementation of Video Adjustment Based on MMX in DirectDraw

SONG Fang-wei^{1,2}, LIU Yong¹, NIE Shi-liang², WANG Neng¹

(1. Dept. of Military Electronic, No. 58 Research Institute of China Ordnance Industries, Mianyang 621000, China;

2. School of Information Engineering, Southwest University of Science & Technology, Mianyang 621010, China)

Abstract: Aiming at the shortage of the interface function for video adjustment in DirectDraw, used multimedia enhance instruction set (MMX) to realize video adjustment interface. Based on the general DirectDraw workflows, double buffer principle and realization of video display in DirectDraw, analyze the principle of DirectDraw player. Introduce the workflow of MMX and make an experimental analysis. The experiments results prove the method has high efficiency and high speed, and the video is easy and smooth.

Keywords: DirectDraw; MMX; Video adjustment

0 引言

DirectX 是微软推出的基于 Windows 操作系统的流媒体应用程序接口函数 API, 主要包括 DirectDraw 和 DirectShow。其中 DirectDraw 主要负责 2D 加速, 实现对显示内存和系统内存的直接操作; 而 DirectShow 是微软公司为简化或减低音/视频开发难而为音视频开发人员定制的开发包、软件库。采用 DirectShow, 虽然能在很短的时间内搭建自己的视频播放系统, 但是由于对显示设备、编/解码算法、音/视频同步等技术的屏蔽, 当需要对这些技术细节进行处理时, 很不灵活; 而 DirectDraw 直接对显示设备进行操作, 可以定位到具体的帧、像素, 因此可灵活地对视频进行处理。但如果采用标准 C 函数来实现海量的数据处理, 会占用大量的系统资源, 其效率也很低。MMX 利用单指令多数数据流 (Single-Instruction Multiple-Data, SIMD) 技术, 开发了很多算法内部蕴藏的并行机制^[1], 使播放系统的效率得以提高。故采用 Intel 公司的 MMX 媒体增强指令集技术来实现视频调节显示功能。

1 DirectDraw 视频播放器原理及设计

1.1 DirectDraw 的一般工作流程

- 1) 创建一个 DirectDraw 对象;
- 2) 设置协作模式;
- 3) 设置显示模式;
- 4) 创建主表面和一个离屏面;
- 5) 往离屏面写入帧数据;
- 6) 通过 Flip (页面翻转) 交换主表面和后台表面, 在这个工作流程中, 最后 2 步完成了显示一帧的工作, 以每秒固定的频率反复进行, 也可按照定时器时间进行, 十分灵活。

1.2 双缓冲显示原理

创建一个主表面, 一个离屏面, 在离屏面中先写入数据, 再通过 Flip 交换到主表面显示出来的方法称为双缓冲。如果不这样做, 直接画到可见的主表面上面, 而省去了离屏面及 Flip 过程, 屏幕上会有明显的闪烁^[2]。

1.3 directdraw 显示实现

由于 YUV 格式的视频在视频压缩、通讯、编码方面广泛使用, 以 YUV4:2:0 格式视频为例实现视频显示和视频调节。DirectDraw 相关函数包括在

收稿日期: 2009-10-03; 修回日期: 2009-12-17

作者简介: 宋方伟 (1983-), 男, 四川人, 在读硕士, 从事音视频处理、嵌入式系统设计研究。

ddraw.h 中:

```

BOOL InitDDraw(void)
{
    // 创建 DirectDraw 对象
    DirectDrawCreate(NULL, &lpDD, NULL);
    // 设置为独占模式
    lpDD->SetCooperativeLevel(hWndCopy,
    DDSCL_EXCLUSIVE | DDSCL_FULLSCREEN);
    // 设置显示模式
    lpDD->SetDisplayMode(ScreenW,ScreenH, BitDepth);
    // 创建主表面
    ZeroMemory(&ddsd, sizeof(ddsd));
    ddsd.dwSize = sizeof(ddsd);
    ddsd.dwFlags = DDSCL_CAPS;
    ddsd.ddsCaps.dwCaps= DDSCAPS_PRIMARYSURFACE;
    lpDD->CreateSurface(&ddsd,&lpDDSPPrimary, NULL);
    // 创建 (YUV4: 2: 0) 格式离屏表面
    ZeroMemory(&ddsd, sizeof(ddsd));
    ddsd.dwSize = sizeof(ddsd);
    ddsd.ddsCaps.dwCaps = DDSCAPS_OFFSCREENPLAIN;
    ddsd.dwFlags=DDSD_CAPS|DDSD_HEIGHT|DDSD_WIDTH|
    DDSD_PIXELFORMAT;
    ddsd.dwWidth = width;
    ddsd.dwHeight = height;
    ddsd.ddpfPixelFormat.dwSize=sizeof(DDPIXELFORMAT);
    ddsd.ddpfPixelFormat.dwFlags=DDPF_FOURCC|DDPF_YUV;
    ddsd.ddpfPixelFormat.dwFourCC = MAKEFOURCC('Y','V',
    '1','2');
    ddsd.ddpfPixelFormat.dwYUVBitCount = 8;
    lpDD->CreateSurface(&ddsd, &lpDDSOffscreen, NULL);
}

```

初始化后把像素数据写到离屏面，再翻转到主屏幕便能够显示。

2 MMX 技术的使用

MMX 技术是对 IA 指令集增强的指令系，可用 SIMD 形式将 64 位寄存器中所有的打包数据元素并行操作。MMX 技术增加了 8 个 64 位的 MMX 寄存器 (MM0-MM7)，另外还有 EAX、EBI、ECX、EDX 4 个寄存器用于 MMX 指令。

在进行视频显示时，通常需要加入调节视频亮度、对比度以及让视频旋转等功能。虽然 Directdraw 提供了 IDirectDrawColorControl 和 IDirectDrawGammaControl 接口实现色彩调节，使用 DDBLTFX 结构的 dwRotationAngle 成员作为页面的旋转角度 (specified in 1/100th of a degree) 来完成显示旋转。但是在实践中发现需要显卡支持，而且支持这些接口的显卡很难找到。由于 directdraw 十分靠近底层，可以处理帧和像素。故使用纯 C 语言实现单帧亮度调节：

```

for (i=0; i<h*w;i++)
*(lpy+i)=*(lpy+i)+a;

```

其中，lpy 为指向 Y 分量数据的指针，a 为变化系数，很明显，一般的操作要进行 h*w 次运算。

采用嵌入式 MMX 汇编指令来实现，代码如下：

```

__int64 c=a;
//构造变化系数
for(int l=0;l<8;l++){
    c=c<<8;
    c|=a;} __asm {
    emms
    mov esi, pin
    mov edi, pout
    mov ecx, loop

movq mm1, c;
start_loop:

    movq mm0,[esi]
    paddusb mm0,mm1
    movq [edi],mm0
    add esi,8
    add edi,8
    dec ecx
    jnz start_loop
    emms
}
}

```

pin 为指向亮度分量输入指针，pout 为亮度分量变化后输出指针，a 为亮度增加系数。可以看出，MMX 一次打包 8 个字节，合并到一个寄存器中进行运算。

由于对比度的计算一般采用乘法，而 MMX 指令的乘法相对复杂，故巧妙地采用加法来实现对比度 (即亮度分量) 的变化，代码如下：

```

BYTE* pin = lpY;
int b=__int64) abs(contrast);
__asm {
    emms
    mov esi, pin
    mov edi, pout
    mov edx, loop
start_loop:
    movq mm0,[esi]
    mov ecx,b
    movq mm1,[esi]
//完成整数倍的加法运算
loopadd:
    paddusb mm0,mm1
    dec ecx
    jnz loopadd
    dec edx
    mov ecx,edx
    movq [edi],mm0
    add esi,8
}

```

```

    add    edi,8
    dec    ecx
    jnz    start_loopc
    emms
    }
    lpY=pcout;
}

```

3 试验结果

在奔腾双核 1.73 GHz, 2G 内存, 集成显卡 intel mobile 945 显卡的笔记本上进行试验。基于 Visual C++和 DirectDraw 设计出播放器。通过观察单独播放视频, 选择 c 调节亮度, 选择 MMX 调节亮度等, 得到下面实验结果数据, 如表 1。

试验证明了 DirectDraw 技术在进行视频显示时的灵活性和接近底层硬件的特性。MMX 得益于 IA 的“SIMD”结构, 把 8 个字节打包到 64 位的寄存器进行运算, 使得每次操作 8 个字节, 并且可在一个时钟周期内完成 2 条指令。

(上接第 90 页)

通过如下 2 种方式得到这些数据:

http_request.responseText: 以文本字符串的方式返回服务器的响应

http_request.responseXML: 以 XMLDocument 对象方式返回响应。处理 XMLDocument 对象可以用 JavaScript DOM 函数。

根据以上步骤, 可实现一个简单 AJAX 的 Web 应用。通过分析, 可归纳出基于 AJAX 的可用的 Web 开发框架, 供重复调用。完整的框架代码如下:

```

<script language="javascript">
var http_request = false;
function send_request(url) { //初始化、指定处理函数、发送请求的函数
http_request = false;
//开始初始化XMLHttpRequest 对象
if(window.XMLHttpRequest) { //Mozilla 浏览器
http_request = new XMLHttpRequest();
if (http_request.overrideMimeType) { //设置MiME 类别
http_request.overrideMimeType("text/xml");
}
} else if (window.ActiveXObject) { // IE 浏览器
try {http_request = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
try {http_request = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e) {}}
if (!http_request) { // 异常, 创建对象实例失败
window.alert("不能创建XMLHttpRequest 对象实例。");

```

表 1 实验结果数据

试验项	播放	亮度调节 (c)	亮度调节 mmx	对比度 (c)	对比度 mmx
cpu 占用率	8%	25%	11%	72%	18%
单帧处理时间		27 μs	5 μs	51 μs	9 μs

4 结束语

采用 MMX 指令实现 DirectDraw 视频调节接口, 其运行速度快、视频流畅、效率高, 能为多媒体数据处理提供有效帮助。

参考文献:

[1] Intel Corporation. Intel Architecture Software Developer's Manual Volume 1: Basic Architecture[Z]. Order Number 243190. USA: Intel Corporation, 1997.
[2] Microsoft Corporation. DirectX 9.0 Programmer's Reference[S]. USA: Microsoft Corporation, 2002.
[3] 骆云志, 刘治红. 视频监控技术发展综述[J]. 兵工自动化, 2009, 28(1): 1-3.

```

return false;}
http_request.onreadystatechange = processRequest;
//确定发送请求的方式和URL 以及是否同步执行下段代码
http_request.open("GET", url, true);
http_request.send(null);}
// 处理返回信息的函数
function processRequest() {
if (http_request.readyState == 4) { // 判断对象状态
if (http_request.status == 200) { // 信息已经成功返回, 开始处理信息
alert(http_request.responseText);
} else { //页面不正常
alert("您所请求的页面有异常。");}}
}
</script>

```

3 结束语

AJAX 技术改变了 Web 应用的交互方式。通过利用客户端闲置的处理能力, 把一些以前需要通过服务器完成的任务转嫁到客户端, 减轻了服务器的负担, 提高了客户端的响应速度, 缩短了用户等待时间, 改善了用户体验。

参考文献:

[1] 孙卫琴. 精通 STRUTS[M]. 北京: 电子工业出版社, 2004.
[2] 阎宏. Java 与模式[M]. 北京: 电子工业出版社, 2002.
[3] 宁波. J2EE 结合UML在企业级系统中的应用[J]. 计算机工程与科学, 2004, 26(13): 82-86.
[4] 孙卫琴. 精通 Struts: 基于 MVC 的 Java Web 设计与开发 [M]. 北京: 电子工业出版社, 2004.