

DirectDraw 技术 在多光谱图像实时显示中的应用*

王 沛 袁晓兵

(中国科学院长春光学精密机械研究所 长春 130022)

摘 要 在高分辨率成像光谱仪的工作中,要求在不影响诸如采集、存储等优先级高的工作条件下,实时显示多光谱图像,用传统的 GDI 是无法实现的。文中提出的 DirectDraw 理论和实现方法,可实时显示多光谱图像并已应用于 863-2 高分辨率成像光谱仪系统中。

关键词 DirectDraw 硬件抽象层 硬件仿真层 DMA

中图分类号 TP752.2 **文献标识码** A

1 引 言

虽然现在很多应用软件几乎都毫不例外地转到了 Windows 平台上,但对于需要实时图像处理来说,由于 Windows 系统的特殊性,这种过度却很难,Windows 接管了所有的系统资源,对设备的操作实际上变成了对 Windows 功能的调用。尽管这样就可以很容易地实现设备无关性,无需重写代码,编制的软件就可以轻松地在不同软硬件环境下运行。不过,在得到这种便利的同时也得付出代价,调用 Windows 的 API(Application Program Interface 应用程序接口)函数比直接操作硬件要多花费很多时间。对于一般的应用程序来说这种影响可能还不是太明显,特别是在目前机器性能飞速提高的情况下更是如此。但对于 863-2 高分辨率成像光谱仪任务中实时图像处理来说,对大量的图形、图像数据处理时如果每次都去调用 API 函数,当然会很慢以至于影响软件的实际应用。而在 Windows 环境下强行通过直接操作硬件(如端口读写)来提高软件运行的效率又容易引起系统崩溃(再高明的程序员也不容易对操作系统的内部结构有透彻的了解),这就给 863-2 光谱图像的实时处理带来了很大的困难。

在高分辨率成像光谱仪的工作中,要求对两路探测器来的多光谱图像进行实时采集、存储、显示,而显示系统既不能影响优先级高的采集、存储的工作;又要尽量节省系统资源,同时实时显示光谱图像,用传统的 GDI(Graphic Device Interface 图形设备接口)是无法实现的。

* 国家 863-2 高技术计划资助项目

收稿日期:1999-02-26

修稿日期:1999-05-04

Microsoft DirectX 有一整套出色的 API, 为开发多媒体应用提供了技术支持和开发工具, 开发者可以通过这些应用程序接口设计高性能、实时的应用程序。我们利用其中的 DirectDraw 技术, 实现了多光谱图像的实时显示。

2 DirectDraw 理论

2.1 DirectDraw 基本概念

利用 DirectDraw 技术, 实现了 863- 2 成像光谱仪高性能的多光谱图像实时显示。

DirectDraw 是 DirectX SDK 的主要部分之一, 其概念类似于“直接写屏”。它允许应用程序直接访问硬件和显示内存, 直接对显示内存操作, 支持硬件位块传输、硬件覆盖、表面翻转, 并且保持同目前的基于 Windows 的应用程序和驱动程序兼容, 通过该技术可加速图像的显示。

DirectDraw 可以运行在 Windows95/NT4.0 和以后的版本中, 是一种软件接口, 它除了能直接对显示设备存取外, 还保持同 Windows 图形设备接口 GDI 兼容。DirectDraw 提供了一种设备无关性的方法, 使得基于 Windows 的应用软件能直接获取显示设备的特性。

2.2 DirectDraw 的主要功能

DirectDraw 提供了以下一些以前只有通过特定显示设备编写专门的代码才能获得的好处:

a. DirectDraw 能工作于各种各样的显示设备, 从简单的 SVGA 显示器到能够提供剪切、拉伸和非 RGB 格式支持的高级显示设备。DirectDraw 提供的对显示内存的设备无关性的访问能让你很容易地管理显示内存。应用程序只需要识别一些基本的设备属性, 例如 RGB 和 YUV 格式的颜色。不需要调用特殊的过程来使用位块传输或操纵调色板寄存器。使用 DirectDraw, 可以很容易地操作显示内存, 充分利用不同类型显示设备的位块传输和颜色解压功能, 而不需要依赖于特定的硬件。

b. DirectDraw 运行在 Windows 95 上, 能够利用 32 位内存的优越性和操作系统提供的“平坦”内存模型。DirectDraw 将系统内存和视频内存作为大块存储而不是一小段一小段地使用。DirectDraw 还能提供表征真实视频内存的绘图表面, 一个表面表征了一块线性内存区域, 这就意味着当你使用 DirectDraw 时, 能够直接向视频内存写数据, 使得图象的显示速度足够快, 并且寻址更加容易。表面通常都驻留在显示内存中, 当然也可以存在于系统内存中。除非明确指定, DirectDraw 会把表面放置于能够获得最佳性能的位置, 并利用显示卡上的特殊处理器, 不仅可以加快任务的执行速度, 还可以同系统 CPU 并行处理一些任务。

c. DirectDraw 可在全屏模式下应用多个后台缓冲区的页翻转。

d. 支持窗口模式和全屏模式下的剪切功能。

e. 支持三维 Z 缓冲区。

f. 支持由硬件参与的有深度信息的覆盖操作。

g. 提供对图象拉伸的硬件的存取。

h. 同时使用显示设备的标准和增强内存区域。

i. 动态调色板、独占式的硬件访问、分辨率切换等。

将这些特性结合在一起, 就可以较容易地编制出性能超过基于 GDI 的标准 Windows 应用程序。

2.3 DirectDraw 的硬件抽象层和硬件仿真层理论

DirectDraw 和其它的 DirectX 部分一样, 能发挥硬件的最优性能, 并且当硬件支持无效时, 提供软件模仿来实现更多功能。

DirectDraw 通过硬件抽象层(Hardware Abstraction Layer, 即 HAL)实现与设备无关, 硬件抽象层是一个由厂商提供的特殊设备接口, 使 DirectDraw 直接工作在显示硬件上, 应用程序使用 DirectDraw 的一整套相容的接口和方法来显示图形。设备厂商在一个 Windows 下的 16 位码和 32 位码结合层上实现硬件抽象层, 在 Windows NT 下, 硬件抽象层通常执行 32 位码。DirectDraw 的硬件抽象层提供了统一的接口, 通过该接口, 程序可以直接在显示内存或视频内存中工作, 获取硬件的最佳性能。硬件抽象层可以是显示设备的一部分, 也可以是一个分离的动态链接库通过设备制造者定义的专用接口和显示设备通信。

DirectDraw 对视频硬件的性能进行估计, 只要可能就会使用硬件提供的特定性能。例如, 如果视频卡支持位块传输, DirectDraw 就会把位块传输委托给视频卡, CPU 不参与位块传输的处理, 这就大大提高了程序运行的性能。另外, DirectDraw 提供了硬件仿真层 HEL (Hardware Emulation Layer), 使得在某些硬件不存在时可以用软件仿真来支持本应该由这些硬件提供的特性。和硬件抽象层一样, 硬件仿真层对应用程序来说, 大部分重要功能的支持是透明的。

下图显示了 DirectDraw、图形设备接口 GDI、硬件抽象层 HAL、硬件仿真层 HEL 和硬件之间的关系。

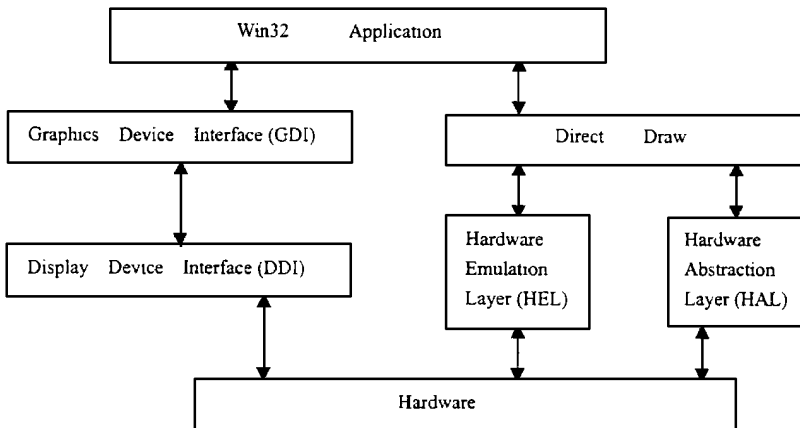


Fig. 1 The relationships among DirectDraw、GDI、HAL、HEL and the hardware

如图中所示, DirectDraw 同 GDI 是并列关系, 两者都通过设备相关抽象层访问硬件, 同 GDI 不同的是, DirectDraw 只要有可能就会充分利用硬件提供的特有功能, 如果硬件不支持某一特性, DirectDraw 还会尝试用 HEL 来仿真。显然, 软件模仿不可能与硬件提供的功能实现相等, 所以在实时图像处理中尽可能调用 HAL 而不是 HEL 以提高处理速度。DirectDraw 也

能以设备上下文(Device Context)的形式提供表面内存,使你能够使用 GDI 函数来处理表面对象。GDI 的缺点是它并不是为高性能的多媒体软件设计的,它主要用于商业软件如字处理和电子表格软件, GDI 提供了对系统内存中的视频缓冲区的访问,但不提供对视频内存的访问,所以在实时图象处理中, GDI 速度显得太慢了。

2.4 DirectDraw 的高级特性- 直接内存访问 DMA

有些显示设备能够在系统内存表面支持位块传输操作(或其它操作),这些操作就是直接内存访问 DMA(Direct Memory Access),可以利用 DMA 的支持来加速一定的操作。例如,在这种设备上可以从系统内存位块传输到视频内存上,同时处理器还准备处理下一帧的画面。

在使用 DMA 操作之前,必须检测设备是否支持 DMA 及支持的程度。步骤是先用 IDirectDraw2::GetCaps 方法来获取设备的能力,然后再查看 DDCAPS 结构中的 dwCaps 成员是否含有 DDCAPS_CANBLTSYSMEM 标志,若是则表明设备支持 DMA。

如果设备支持 DMA,还需要知道驱动程序是如何支持的,这就需要查看有关的结构成员,这些成员包含了有关系统内存——视频内存、视频内存——系统内存和系统内存——系统内存之间位块传输操作的信息。DDCAPS 结构的几个成员提供了这些信息,如下表所示:

Table 1 The members of DDCAPS structure

System to video	Video to system	System to system
dwSVBCaps	dwVSBCaps	dwSSBCaps
dwSVBCKeyCaps	dwVSBCKeyCaps	dwSSBCKeyCaps
dwSVBFXCaps	dwVSBFXCaps	dwSSBFXCaps
dwSVBRops	dwVSBROps	dwSSBROps

这些成员最关键的特性是对异步 DMA 位块传输操作的支持。如果驱动程序支持表面间的异步 DMA 位块传输操作,DDCAPS_BLTQUEUE 标志将会设在 dwSVBCaps、dwVSBCaps、和 dwSSBCaps 成员中,若该标志不存在,就表明驱动程序不支持异步 DMA 位块传输操作。

系统内存到视频内存之间的 SRCCOPY 传输是硬件支持的最常见的位块传输操作,因此,对这种操作最典型的应用是从系统表面内存将许多纹理移动到视频内存,系统内存到视频内存的 DMA 传输大约同处理器控制的传输速度一样快,但具有更大的优点,因为这些操作可以同主机处理器并行运行。

硬件传输使用物理内存地址而不是虚拟内存地址,某些设备驱动程序要求提供物理内存地址。该机制是通过 IDirectDrawSurface3::PageLock 方法的调用完成的,如果设备驱动程序不要求页锁定,在用 IDirectDraw2::GetCaps 方法获取硬件的能力时,DDCAPS2-NOPAGELOCKREQUIRED 标志将被设定。

页锁定一个表面可以防止系统将表面的物理内存再做它用,并且保持该表面的物理内存不变,直到调用 IDirectDrawSurface3::PageUnlock 方法为止。如果设备驱动程序请求页锁定,DirectDraw 只允许 DMA 操作在应用程序已经页锁定了的系统内存表面上,如果在这种情况下没有调用 IDirectDrawSurface3::PageLock,DirectDraw 将用软件仿真执行传输工作。需要注意的是,锁定的表面多了之后将会大大降低 Windows 运行的效率,因此,最好只在全屏独占模式下处理较多数量的系统内存,并且当应用程序最小化时能够对这些表面进行解锁。当然,

应用程序恢复后,应该重新对系统内存表面页锁定。

3 应用及结论

综上所述可以看出, DirectDraw 技术的最终目的就是为了让 Windows 图形系统下实现快速图像显示的能力。下面是用 DirectDraw 进行的系统测试,系统配置为 Pentium 300, 显示卡为 Intel i740(8M), 系统平台为 Windows 98, 结果如下表(每次测试结果略有不同,取平均值)。

Table 2 The experimental results

system to system dword copy	110.58 MB/s
system to video dword copy	153.85 MB/s
system to system byte copy	68.37 MB/s
system to video byte copy	80.48 MB/s
system dword fill	189.21 MB/s
video dword fill	251.13 MB/s
system byte fill	128.01 MB/s
video byte fill	251.89 MB/s
system vertical byte fill	11.16 MB/s
video vertical byte fill	17.07 MB/s
320× 240 frame rate	66.4 FPS
640× 480 frame rate	56.4 FPS
320× 240 stretched to 640× 480	56.2 FPS
800× 600 frame rate	52.6 FPS
1024× 768 frame rate	44.2 FPS
1280× 1024 frame rate	40.2 FPS

以上结论已经应用于 863-2 成像光谱仪图像实时采集、存储、显示系统中,采集、存储系统工作于 Windows Kernel 模式下,为节约系统资源,实时显示部分工作于 User 模式下,整个系统工作框图如下:

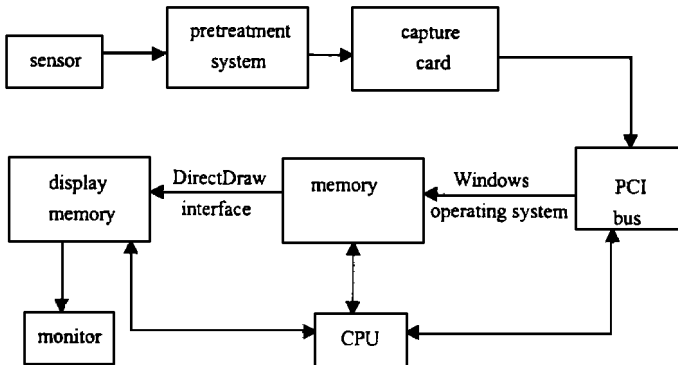


Fig. 2 The real time capturing, storing and displaying system of multispectral images

在 863-2 高分辨率成像光谱仪项目中, 128 波段的图像信号分两路采集至主机内存, 大小为 $256 \times 128 \times 16 \times 60 = 3.75 \text{MB/s}$, 实验表明, 用 DirectDraw 编制的图像实时显示系统在整个

系统中工作正常,在不影响其它如采集、存储等优先级高的工作的条件下 512×256 bit 的光谱图像显示速率可控制在 60 FPS,很好满足实时性的要求。

参 考 文 献

- 1 Stan Trujillo. High Performance Windows Graphics Programming. Published by The Coriolis Group, Inc. 1998
- 2 Microsoft corp. Direct X 6.0 SDK documentation. Microsoft corp, 1998
- 3 曾志,李舒平等. Win32 高级图形编程技术. 成都: 电子科技大学出版社, 1998

Application of DirectDraw in Real Time Displaying Multi-spectral Images

WANG Pei, YUAN Xiao-Bing

(Changchun Institute of Optics and Fine Mechanics,
Chinese Academy of Sciences, Changchun 130022)

Abstract

When the high resolution imaging spectrometer (HRIS) works, the display of multi-spectral images in real time is required without affecting such prior working conditions as capture and storage, which can not be realized by traditional GDI. This article puts forward DirectDraw theory and gives its realized method, which can display multi-spectral images in real time and has been applied to No. 863-2 HRIS system.

Key words: Direct Draw, Hardware Abstraction Layer, Hardware Emulation Layer, DMA

王 沛 女, 1970 年 10 月出生, 1992 年 7 月毕业于浙江大学信息与电子工程学系, 1997 年 3 月于中国科学院长春光机所获得硕士学位, 目前在长春光机所攻读博士学位。所从事的工作和研究方向为计算机图像处理。