

文章编号:1001-9081(2007)05-1171-03

## 基于 PC 的 OpenGL 跨平台实现及性能研究

朱奕杰

(哈尔滨工程大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

(zhuyijie@sina.com)

**摘要:**针对 OpenGL 独立于操作系统和窗口系统的特性,分析了 OpenGL 基于 PC 的工作原理,包括泛型实现、硬件实现和 OpenGL 跨平台实现。使用 OpenGL 性能特征小组的 SPECViewperf 9.0 对 OpenGL 在 Windows 和 Linux 操作平台上的实现性能进行了测评和对比,并给出相关数据和图表。结果显示,OpenGL 运行在硬件模式时会获得更好的性能。在 PC 平台上,OpenGL 的实现性能受操作系统和窗口系统影响。测评方法和结果对选择 OpenGL 操作平台具有一定参考价值。

**关键词:**OpenGL;性能;测评;SPECViewperf

**中图分类号:**TP302.7 **文献标识码:**A

## Research of OpenGL implementations and performance based on PC platforms

ZHU Yi-jie

(College of Computer Science & Technology, Harbin Engineering University, Harbin Heilongjiang 150001, China)

**Abstract:** OpenGL is operation system and window system independent. The approach of OpenGL implementations based on PC, including the software implementation, Hardware-accelerated implementation, and the implementation across different platforms was analyzed. Furthermore, the performance of OpenGL implementations based on Windows operation systems and Linux distributions was measured with OpenGL Performance Characterization group's SPECviewperf 9.0 benchmark and some performance figures were given. The results indicate that Hardware - accelerated OpenGL implementation can attain optimized performance. And for PC platform, the performance of OpenGL implementations was affected by operation system and window system. Benchmarking and Performance Evaluation help us to select a platform for optimized performance of OpenGL.

**Key words:** OpenGL; performance; benchmark; SPECViewperf

### 0 引言

OpenGL 作为开发灵活的、交互的 2D 和 3D 图形应用程序的一个主要环境,目前已经成为工业界最广泛使用和支持的 2D 和 3D 图形应用程序接口(API),它是真正意义上开放的、跨平台的图形标准<sup>[1]</sup>。随着并不昂贵但性能卓越的图形硬件的普及,以及原来只在图形工作站上才具有的 OpenGL 硬件实现技术在 PC 上的应用,OpenGL 在 PC 上得到了广泛的运用,例如在 Windows 系统上应用于机器人模拟<sup>[2]</sup>和虚拟地形漫游系统<sup>[3]</sup>,以及在 Linux 环境下模型直升机飞行过程的实时 3D 仿真<sup>[4]</sup>和地震数据的三维可视化<sup>[5]</sup>中的应用。同时其性能成为至关重要的问题。目前对于 OpenGL 性能的主要研究可分为两个方面:一是针对计算机 3D 图形卡、CPU 等“硬件层”的研究,例如 J. W. Sheaffer, D. Luebke, K. Skadron 开发的 GPU 模拟工具 Qsilver<sup>[6]</sup>,以及 Chris J. Thompson, Sahngyun, Hahn Mark Oskin 对图形硬件结构的研究<sup>[7]</sup>;二是对应用软件的“应用软件层”的研究,例如罗朔锋,李雪耀,熊新平等在高性能面向对象场景图系统对 OpenGL 图形性能的优化<sup>[8]</sup>。然而,OpenGL 在跨平台的具体实现时,其性能除了与“硬件层”和“应用软件层”相关外,由于 OpenGL 没有附属任何窗口函数,OpenGL 必须要与具体的操作系统和窗口系统相结合才能执行<sup>[9]</sup>,使得 OpenGL 的性能又与具体操作系统及窗口系统这一“操作平台层”密切相关。

针对这一问题,本文对基于 PC 的 OpenGL 跨平台实现性能作了研究,分析了 OpenGL 跨平台实现的原理,详细论述了使用 SPECViewperf 9.0 对 OpenGL 在 Windows 和 Linux 上实现性能的测评,分析和比较了 PC 上不同操作平台对 OpenGL 性能的影响。

### 1 OpenGL 工作原理

#### 1.1 实现模式

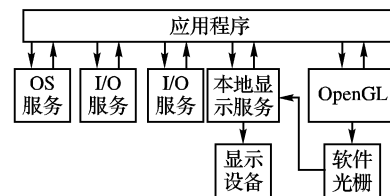


图 1 泛型实现中 OpenGL 在一个典型应用程序中的位置

OpenGL 的实现有两种模式:泛型实现和硬件实现。泛型实现即软件实现,也就是对 OpenGL 函数调用做出响应创建三维图像的软件函数库。图 1 显示当一个应用程序运行时 OpenGL 的泛型实现所占据的典型位置。在不同的操作系统上想要在屏幕上创建输出的应用程序通常会调用一种本地显示服务。OpenGL 的硬件实现又称为加速实现,通常采用图形卡驱动程序的形式。图 2 显示了它与应用程序的关系。OpenGL 调用将传递给硬件驱动。这个驱动程序并不把它的

输出传递给本地显示服务以进行显示,而是直接与图形显示硬件进行通信<sup>[10]</sup>。需要说明的是,OpenGL的硬件实现中通常也包含了 OpenGL 的软件实现,这在图 2 中并未明确反映出。

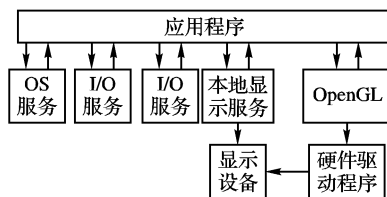


图 2 硬件实现中 OpenGL 在一个典型应用程序中的位置

## 1.2 OpenGL 在 PC 上的实现

OpenGL 是操作系统独立的和窗口系统独立的。为了将它结合到各种窗口系统中,需要利用附加库,为此每个窗口系统有它自己的特定的库和函数。在 Linux 上,X Windows 系统和 OpenGL 之间的“桥梁”是 GLX;在微软的 Windows 上则是 WGL。对于 Windows 操作系统,Windows NT 3.5 和 Windows 95 (Service Release 2 以后)之后的每个版本都提供了 OpenGL 的实现。Windows 2000 和 Windows XP 也包含了对 OpenGL 的支持。在屏幕上创建输出的 Windows 应用程序通常会调用一种叫做图形设备接口 (GDI) 的 Windows API。一般来说,如果安装系统时使用操作系统自带的显卡驱动程序,那么 OpenGL 程序就会运行在泛型实现模式下,而安装显卡驱动程序之后,OpenGL 就会运行在硬件实现模式下。

Linux 上通常使用直接渲染结构 (Direct Rendering Infrastructure, DRI) 提供加速的 OpenGL 实现。Mesa 是一个开源的 OpenGL 实现,通常它是作为 DRI 的核心与 DRI 一起完成 OpenGL 硬件实现<sup>[11]</sup>,而当 3D 图形硬件不在 Mesa 支持或使用的范围内时,Mesa 完成 OpenGL 的泛型实现。一般来说,Linux 上 3D 图形硬件驱动是基于 DRI 的,但是也有一些例外,比如本文测试平台中使用的 3D 图形硬件 NVIDIA 的驱动就不是基于 DRI (安装该驱动程序会在 X Window 配置文件中去掉“载入 DRI”一项)。它提供了自己的非开源驱动程序,并为 Linux 与 Windows 使用一致的驱动代码。其驱动方式在 Linux 上为内核模式驱动,在 Windows 上是以 VXD 最小端口方式驱动。为了获得 OpenGL 的硬件实现,本文在 Windows 各版本和 Linux 各发行版中均安装 NVIDIA 最新显卡驱动程序 (作者写稿时,Windows 最新版本是 91.47, Linux 最新版本为 1.0-9625 beta)。

## 2 测试平台

### 2.1 硬件平台

本文测试计算机 (I),代表了中低水平的 PC 性能,其配置如下:CPU P4 3.00GHz,内存 512MB,显卡 NVIDIA GeForce FX 5200。

在计算机高速发展的今天,双核技术已经成为 PC 发展的方向。为了更加全面的比较 OpenGL 在相对高端性能的 PC 上实现性能的不同,本文使用“双核”测试计算机 (II),其配置如下:CPU PD 2.80GHz,内存 1GB,显卡 NVIDIA GeForce 7300 LE。

### 2.2 操作系统

在 Windows 平台上本文选用操作系统如下:

- (1) Windows 2000 Professional
- (2) Windows 2000 Advanced Server

(3) Windows XP Professional

(4) Windows Server 2003, Enterprise Edition

在下文中,Windows XP Professional 和 Windows Server 2003, Enterprise Edition 将会分别用 Windows XP 和 Windows Server 2003 代替。

Linux 作为 OpenGL 实现的一个主要操作系统平台,除了是免费分发之外,还因其多功能性、可改写性和健壮性,成为 UNIX 和微软操作系统的主要替代者<sup>[12]</sup>。作为 Linux 操作系统核心的内核也由最初的 0.01 版本发展到目前的 2.6 版本 (2003 年 12 月发布)。而基于内核之上的 Linux 发行版有上百种,本文选择其中 3 个有代表性的 Linux 发行版: Fedora Core 5、Ubuntu 6.06 和 RedFlag 5.0。

## 3 性能分析程序 SPECViewperf

### 3.1 SPECViewperf 简介

SPECViewperf 是 OpenGL 性能特征组织 (OpenGL Performance Characterization, SPECopC) 发布的一个 OpenGL 性能测评程序,它由 IBM 用 C 语言编写开发,后经 SGI、Digital、3Dlabs 和其他 SPECopC 项目小组的贡献使之不断完善。目前,该程序能够运行于绝大多数操作系统,包括 UNIX, Windows 系列以及 OS/2<sup>[13]</sup>。本文使用 SPECViewperf 9.0 对 OpenGL 性能进行测评。

SPECViewperf 9.0 对 OpenGL 的测评数据源 (称之为 Viewset) 并不是由 SPECopC 项目小组开发的,而是来自于独立软件提供商 (Independent Software Vendor, ISV) 本身。目前,共有 9 个 SPECopC Viewset: 3DSMAX, CATIA, ENSIGHT, LIGHT, MAYA, PROE, SW, UGNX 和 TCVIS。

### 3.2 SPECViewperf 测评基准和方法

SPECViewperf 是通过测试每秒帧数 (Frames Per Second, FPS) 反映 OpenGL 性能。其他相关信息如测试的操作系统版本,所有的渲染状态,建立显示列表时间 (如果使用了的话) 以及使用的数据源等都输出在标准化的报告中。以 3DSMAX Viewset 为例,3DSMAX Viewset 当中包含了 14 个不同的测试包,每个测试包具有一个权重,由公式 (1) 计算得到一个加权几何图形平均值:

$$\sum_{i=1}^n (\text{frames/second}_i)^{w_i} \quad (1)$$

式中:

$n$  — Viewset 中独立测试包总数。

$w$  — 每个独立测试包的权重值。

## 4 性能测试

以下将使用 SPECViewperf 9.0 对 OpenGL 在 Windows 和 Linux 平台上的实现性能做出综合测评和分析,测评过程中测试窗口大小均为宽 1280 高 1024。

### 4.1 OpenGL 不同实现模式性能比较

OpenGL 运行在不同实现模式时实现环境有所不同,在 Windows 上主要不同如表 1 所示。OpenGL 版本方面,Windows 的 OpenGL 实现为 1.1 版本,事实上,Windows 对于 OpenGL 的支持在 1.1 版本之后就再也没有更新过。泛型实现模式下 OpenGL 的本地显示服务是 GDI,而在硬件实现模式下显卡驱动提供了对 OpenGL 最新的支持,对 OpenGL 的调用则直接由硬件驱动来完成。

在 Linux 上,与 Windows 对 OpenGL 支持不多不同,Mesa 近几年来一直紧随 OpenGL 逐渐增加特性(如表 2 所示),Mesa 6. x 已经支持 OpenGL 1.5。当 OpenGL 运行在泛型模式时,Mesa 为间接渲染状态,在硬件实现模式下则为直接渲染状态。然而正如表 3 和表 4 所示,不论是在 Windows 还是在 Linux 上,OpenGL 运行在硬件模式时的性能都要优于泛型模式时几倍甚至更多。

表 1 Windows 实现环境主要不同比较

| 不同之处                | 泛型实现        | 硬件实现                     |
|---------------------|-------------|--------------------------|
| OpenGL Vendor       | MICROSOFT   | NVIDIA                   |
| OpenGL Version      | 1.1.0       | 2.0.3                    |
| OpenGL Renderer     | GDI Generic | GeForce FX 5200/AGP/SSE2 |
| OpenGLClientVendor  | MICROSOFT   | NVIDIA                   |
| OpenGLClientVersion | 1.1.0       | 2.0.3                    |
| Direct Rendering    | False       | False                    |

表 2 Linux 实现环境主要不同比较

| 不同之处                | 泛型实现                 | 硬件实现                       |
|---------------------|----------------------|----------------------------|
| OpenGL Vendor       | MESA                 | NVIDIA                     |
| OpenGL Version      | 1.2<br>(1.5 Mesa)    | 2.1.0 NVIDIA<br>96.25      |
| OpenGL Renderer     | Mesa GLX<br>Indirect | unknown board<br>/AGP/SSE2 |
| OpenGL ClientVendor | SGI                  | NVIDIA                     |
| OpenGLClientVersion | 1.4                  | 1.4                        |
| Direct Rendering    | False                | True                       |
| GLX Server Version  | 1.2                  | 1.3                        |

表 3 Windows XP 上 OpenGL 运行在不同实现模式性能比较

|      | 3DSMAX | CATIA  | ENSIGHT | LIGHT  | MAYA  | PROE   | SW     | UGNX   | TCVIS  |
|------|--------|--------|---------|--------|-------|--------|--------|--------|--------|
| 泛型实现 | 0.4642 | 0.5193 | 0.4276  | 0.8446 | 1.168 | 0.4028 | 0.7448 | 0.0388 | 0.1987 |
| 硬件实现 | 2.605  | 6.025  | 1.358   | 6.08   | 8.199 | 4.291  | 4.33   | 0.1955 | 1.082  |

表 4 Fedora 5 上 OpenGL 运行在不同实现模式下性能比较

|      | 3DSMAX | CATIA  | ENSIGHT | LIGHT  | MAYA   | PROE   | SW     | UGNX   | TCVIS  |
|------|--------|--------|---------|--------|--------|--------|--------|--------|--------|
| 泛型实现 | 0.3864 | 0.2631 | 0.3026  | 0.5759 | 0.6996 | 0.1791 | 0.3163 | 0.0120 | 0.0824 |
| 硬件实现 | 2.609  | 5.615  | 2.391   | 5.504  | 7.939  | 4.064  | 3.992  | 0.7092 | 1.045  |

表 5 Windows 平台上 OpenGL 性能比较

|                             | 3DSMAX | CATIA | ENSIGHT | LIGHT | MAYA  | PROE  | SW    | UGNX   | TCVIS  |
|-----------------------------|--------|-------|---------|-------|-------|-------|-------|--------|--------|
| Windows Server 2003         | 1.475  | 1.852 | 0.4947  | 1.599 | 2.283 | 1.172 | 2.389 | 0.1056 | 0.5284 |
| Windows XP                  | 2.605  | 6.025 | 1.358   | 6.08  | 8.199 | 4.291 | 4.33  | 0.1955 | 1.082  |
| Windows 2000 Advance Server | 2.592  | 5.149 | 1.199   | 5.248 | 7.697 | 3.925 | 4.23  | 0.151  | 1.082  |
| Windows 2000 Professional   | 2.59   | 5.149 | 1.224   | 5.35  | 7.887 | 3.993 | 4.343 | 0.1508 | 1.082  |

表 6 Linux 平台上 OpenGL 性能比较

|             | 3DSMAX | CATIA | ENSIGHT | LIGHT | MAYA  | PROE  | SW    | UGNX   | TCVIS |
|-------------|--------|-------|---------|-------|-------|-------|-------|--------|-------|
| Fedora 5    | 2.609  | 5.615 | 2.391   | 5.504 | 7.939 | 4.064 | 3.992 | 0.7092 | 1.045 |
| Ubuntu 6.06 | 2.695  | 5.814 | 2.59    | 5.846 | 8.257 | 4.24  | 4.178 | 0.7414 | 1.095 |
| RedFlag 5.0 | 2.686  | 5.707 | 2.582   | 5.686 | 8.234 | 4.221 | 4.253 | 0.7722 | 1.094 |

表 7 “双核”计算机 Windows 平台上的 OpenGL 性能比较

|                             | 3DSMAX | CATIA | ENSIGHT | LIGHT | MAYA  | PROE  | SW    | UGNX   | TCVIS |
|-----------------------------|--------|-------|---------|-------|-------|-------|-------|--------|-------|
| Windows Server 2003         | 7.081  | 8.119 | 3.387   | 6.918 | 12.29 | 5.725 | 11.28 | 0.7305 | 1.435 |
| Windows XP                  | 7.067  | 8.2   | 3.995   | 6.884 | 12.5  | 5.745 | 11.28 | 0.7858 | 1.433 |
| Windows 2000 Advance Server | 7.218  | 8.167 | 5.092   | 6.999 | 12.41 | 5.757 | 11.29 | 0.7535 | 2.522 |
| Windows 2000 Professional   | 7.216  | 8.157 | 4.668   | 7.304 | 12.4  | 5.757 | 11.28 | 0.7341 | 2.517 |

表 8 “双核”计算机 Windows XP 与 Ubuntu 6.06 的 OpenGL 性能比较

|             | 3DSMAX | CATIA | ENSIGHT | LIGHT | MAYA  | PROE  | SW    | UGNX   | TCVIS |
|-------------|--------|-------|---------|-------|-------|-------|-------|--------|-------|
| Windows XP  | 7.067  | 8.2   | 3.995   | 6.884 | 12.5  | 5.745 | 11.28 | 0.7858 | 1.433 |
| Ubuntu 6.06 | 7.186  | 7.443 | 5.338   | 6.164 | 14.25 | 5.727 | 10.08 | 1.546  | 2.58  |

#### 4.2 OpenGL 的硬件实现性能

下面将针对 OpenGL 的硬件实现性能(以下简称性能)做出测评。在此之前,因为 Linux 各个发行版的内核版本并不都相同,其中 Fedora 5 和 Ubuntu 6.06 内核版本为 2.6.15, RedFlag 5.0 内核版本则为 2.6.9。由于 Linux 内核版本不断更新,新的内核修订了旧内核的 bug,并增加了许多新的特性。为了能够使用这些新特性,并定制一个更高效,更稳定的内核,减少由于内核差别对于测评可能带来的影响,就需要下载最新内核 2.6.18<sup>[14]</sup>,并重新编译安装 Linux 内核(编译安装新内核后,还需要重新安装 Linux 显卡驱动程序)。

表 5 是对 Windows 操作平台上 OpenGL 性能测评的结果。可以看出,Windows XP 上在 CATIA、ENSIGHT、LIGHT、MAYA、PROE 和 UGNX 几项上 OpenGL 性能都最为出色,特别是 CATIA 和 LIGHT 两项,而 OpenGL 在 Windows Server 2003 上较在其他 Windows 版本上性能则有明显的差距。Windows 2000 Professional 和 Windows 2000 Advanced Server 由于版本一样,OpenGL 性能测评结果也基本相同。

相比之下,Linux 操作平台上 OpenGL 性能的差别并不十分明显,如表 6 所示。进一步将 Windows 平台上 OpenGL 性能表现最为突出的 Windows XP 与 Linux 平台上的 Ubuntu 6.06 作一个对比可以看出,虽然 Windows XP 上 OpenGL 性能在 CATIA、LIGHT 和 SW 几项上的测评结果较 Ubuntu 6.06 要稍好一些,但在 ENSIGHT、UGNX 方面却相差比较明显,而在其他各项上的差距则很小。

#### 4.3 “双核”计算机上 OpenGL 性能

SPECViewperf 9.0 在测试过程中使用的 Viewset 代表了相对高端的应用程序,例如 Viewset 中的 CATIA,它所使用的模型包含了总共两百万个顶点。此外,SPECViewperf 9.0 还包括了对光照、显示列表、投影等全面的测评,而完成每一项测评需要大量的系统资源。为了尽量减少由于系统资源的不足对 OpenGL 性能的影响,本文使用测试计算机(II)来进一步认识 OpenGL 跨平台实现的性能。

表 7 是在测试计算机(II)的 Windows 平台上 OpenGL 性能比较。可以看出,在系统资源对于 OpenGL 性能的限制减小后,Windows 平台上各操作系统对于 OpenGL 性能的影响也减少很多,表中显示除了 ENSIGHT 和 TCVIS 两项之外,Windows 各版本上的 OpenGL 性能相差已经非常小。同时,如表 8 所示,Windows XP 与 Ubuntu 6.06 上 OpenGL 性能仅在 3DSMAX 和 PROE 两项上测评结果非常接近,而其他各项测评结果依然是存在较大差异。

### 5 结语

通过对 OpenGL 跨平台实现性能的全面测评可以看出,一方  
(下转第 1186 页)

## 5 分析

运用免疫的思想,对各循环中的进化过程进行分析,从而抽取对蚁群进化较好的蚂蚁的信息段作为疫苗,在进化过程中通过对蚂蚁注射疫苗来加大进化进度,从而提高了搜索效率和求解的质量。基本蚁群算法在进化的后期容易出现信息素急剧积累,很快就出现“早熟”,使得进化无法进行。通过引入免疫算法中的亲和力,设定一定的阈值,对个体相似度过高的个体直接清除;通过将种群按照适应值的大小划分为种群,按照适应值大小排列,分别从这些种群中选择相等数目的个体组成新一代的种群,从而保证了种群的多样性,避免了“早熟”现象。

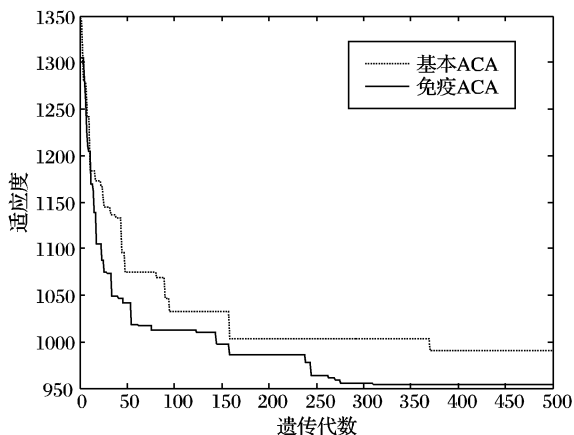


图5 免疫蚁群算法与基本蚁群算法求解 LA06 比较图

## 6 结语

通过研究 Job-shop 调度问题的模型,提出了引入免疫机理来求解该问题的免疫蚁群算法及其具体实现方法。仿真实

验的结果验证了算法的有效性,相对于经典遗传算法求解 Job-shop 调度问题,改进的免疫蚁群算法会以更快更容易收敛到更优的调度结果。

### 参考文献:

- [1] 柴永生,孙树栋,余建军,等. 基于免疫遗传算法的车间动态调度[J]. 机械工程学报, 2005, (10): 23-27.
- [2] DORIGO M. The ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on System, Man, and Cybernetics-part B, 1996, 26(1).
- [3] 赵虎,李睿. 蚂蚁算法在车间作业调度问题中的应用[J]. 计算机工程与应用, 2003, 22(7): 6-8.
- [4] 霍红卫. 算法设计与分析[M]. 西安: 西安电子科技大学出版社, 2005.
- [5] DE CASTRO LN, VON ZUBEN FJ. Artificial immune system: Part I: basic theory and application. Campinas—SR, Brazil. RT-DCA01 [R], 1999.
- [6] 陈雄,李海刚,吴启迪. 基于遗传算法的 Job-shop 调度问题研究[J]. 同济大学学报(自然科学版), 2002, 30(1): 88-91.
- [7] LOVBJERG M, RASMUSSEN TK, KRINK T. Hybrid particle swarm optimization with breeding and subpopulations[A]. Proceeding of third Genetic and Evolutionary computation conference[C]. USA, 2001.
- [8] 李士勇. 蚁群算法及其应用[M]. 黑龙江: 哈尔滨工业大学出版社, 2004.
- [9] 叶志伟,郑肇葆. 蚁群算法中参数  $\alpha$ 、 $\beta$ 、 $\rho$  设置的研究——以 TSP 问题为例[J]. 武汉大学学报(信息科学版), 2004, 29(7): 597-601.
- [10] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.
- [11] 杨阿莉. 一种改进蚁群算法在车间作业调度问题中的研究与应用[J]. 机械与电子, 2005, (4): 9-12.
- [12] 数据的三维可视化[J]. 计算机工程与应用, 2004, (22): 182-184.
- [13] SHEAFFER JW, LUEBKE D, SKADRON K. A flexible simulation framework for graphics architectures[A]. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware [C]. New York: ACM Press, 2004. 85-94.
- [14] THOMPSON CJ, HAHN SY, OSKIN M. Using modern graphics architectures for general-purpose computing: a framework and analysis[A]. Proceedings of the 35th Annual ACM/IEEE International Symposium on Microarchitecture[C]. 2002. 306-317.
- [15] 罗朔锋,李雪耀,熊新平,等. 高性能面向对象场景图系统[J]. 系统仿真学报, 2005, 17(2): 424-428.
- [16] SHREINER D, ANGEL ED, SHREINER V. An interactive introduction to OpenGL programming[A]. ACM SIGGRAPH 2004 Course Notes[C]. New York: ACM Press, 2004. 1-102.
- [17] WRIGHT RS, LIPCHAK B. OpenGL 超级宝典[M]. 第3版. 徐波,译. 北京: 人民邮电出版社, 2005. 22-24, 417-419, 516-519.
- [18] PAUL B. The mesa 3D graphics library[EB/OL]. <http://www.mesa3d.org>, 2006-11-07.
- [19] LinuxOnline! [EB/OL]. <http://www.linux.org>, 2006-11-07.
- [20] SPECViewperf 9.0 [EB/OL]. <http://www.spec.org>, 2006-11-07.
- [21] The Linux kernel archives [EB/OL]. <http://www.kernel.org>, 2006-11-07.

(上接第 1173 页)

面,在相同配置计算机上,相比泛型模式,当 OpenGL 运行在硬件模式时能够发挥更好的性能;另一方面,虽然决定 OpenGL 性能的因素主要是 CPU、图形子系统(包括 GPU、图形总线、驱动)等,但是在 PC 上,操作平台对于 OpenGL 的性能的影响同样是不可忽略的,特别是在运行相对高端的应用程序而计算机系统资源又相对不足时,这种影响体现得更加明显。本文的测评方法和结果对选择 OpenGL 操作平台和衡量 OpenGL 实现性能具有一定参考价值。

本文并未对操作系统和窗口系统可能对 OpenGL 性能发挥产生影响的具体因素以及限制 OpenGL 性能发挥的瓶颈问题做出研究,如何使 OpenGL 在 PC 上发挥出最好的性能需要在今后的学习工作中作进一步深入的研究。

### 参考文献:

- [1] OpenGL overview[EB/OL]. <http://www.opengl.org>, 2006-11-07.
- [2] LING QH, MENG MQH, MEI T, et al. 3D simulation design based on OpenGL for four-legged robot[A]. 2005 IEEE International Conference on Robotics and Biomimetics - ROBIO[C]. 2005. 713-717.
- [3] 李连军,戴金海. 利用 OpenGL 实现卫星系统可视化仿真模型的研究[J]. 计算机仿真, 2004, 21(10): 34-37.
- [4] 李晓强,韩波,李平. Linux 环境下模型直升机飞行过程的实时 3D 仿真[J]. 计算机工程与应用, 2004, (35): 211-214.
- [5] 吴志勤,宁书年,李刚,等. Linux 下基于体绘制算法实现地震