

THE DESIGN AND APPLICATION OF HIGH-RESOLUTION 3D STEREOSCOPIC GRAPHICS DISPLAY ON PC

Duoduo Liao Shiaofen Fang

Department of Computer & Information Science
Purdue University School of Science
723 W. Michigan St., SL280
Indianapolis, IN 46202, U.S.A.

dliao@purdue.edu sfang@cs.iupui.edu

ABSTRACT

This paper describes methods and procedures for real-time rendering of high-resolution 3D stereoscopic graphics using an improved 3D OpenGL accelerator on PC platform. The hardware architecture and software development of the stereoscopic accelerator are first analyzed, followed by in detail description of SDK of stereo display control for Windows NT. Using a special driver and the SDK, the high-resolution 3D stereoscopic graphics is easy to program and control in real-time, with a simple interface to stereo viewing devices. At the end, a few extended stereo PC systems will be discussed. These systems further improve the stereoscopic capacities for low-cost industrial graphics applications.

Keywords: Stereoscopic display, virtual reality, double buffering, quad buffering, stereo synchronization control, stereo SDK.

1. INTRODUCTION

High-resolution stereoscopic display has become one of the most important technologies for embodying immersion in virtual environment [Burdea94]. More and more people would like to develop their applications with this technology based not only on high-end systems, but also on low-cost platforms and personal computers (PCs). Among the existing systems that support stereo functions, some (such as SGI Onyx2) are too expensive to be used widely, and others only display images with very low resolution under interlaced mode or special processing of vertical scanning. For example, Crystal Eyes for PC supports above/below stereoscopic display, *i.e.* the vertical resolution of each of stereo image is only half of the normal image and lots of its details will be lost while enabling the stereo mode. Likewise, the stereo display for SGI's Indy/O2/Indigo2 also belongs to such half mode (*e.g.* 1280x492 pixels per left/right field). The challenge, therefore, is the design and development of a high-resolution, low-cost, real-time

stereoscopic display system for practical and industry applications.

In general, 3D graphics rendering may be performed with or without hardware 3D accelerators. However, a stereo application must draw its scene twice, once for the left-eye view, and once for the right-eye view. Thus, the operations of 3D stereo graphics include 3D graphics rendering and stereo viewing. The design of stereo graphics accelerator consists of two parts: the rendering of general 3D graphics and the stereo image display. Real-time 3D rendering can be achieved using usual OpenGL accelerators with 3D graphics chips such as 3Dlabs' GLINT lines. Therefore, the key component in the design of high-resolution stereo accelerator is to embed stereoscopic display functions to a OpenGL graphics accelerator.

The real-time stereo display system will be described in this paper consists of the high-resolution stereo graphics accelerator, the stereo viewing device, and corresponding software support. The stereo accelerator contains OpenGL graphics chips and stereo control components. The stereo software development kit provides the application

programming interface to the stereo features. The first system was operational in March 1997 (see Figure 7). The subsequent products, AGC-ProV lines, have been used in many low-cost industrial VR applications.

The following sections will describe the cores of these products' hardware design and software development of stereoscopic display. This paper is the first detailed description of the hardware architecture and the SDK of this type of systems.

2. HARDWARE ARCHITECTURE

We are mainly interested in the 3D stereo graphics systems for real-time VR applications on personal computer systems. Since real-time stereo display requires the real-time rendering of stereo pairs, it is important to choose a 3D graphics processor with high speed and lower price. 3Dlabs's GLINT TX was our preference at the beginning of the research in 1996. Based on this processor, 3D stereo graphics accelerator has been designed not only to make full use of its excellent accelerated performance to render 3D graphics with OpenGL [Neider93], but also support the special functions of real-time interactive stereo display.

2.1 Framebuffer for Stereoscopic Display

The framebuffer is a region of memory where the information produced during rasterization is written prior to being displayed. The framebuffer region can hold up to 32Mbytes or more, and there are very few restrictions on the format and size of the individual buffers which make up the video stream[3Dlabs95b]. Each buffer resides at an address in the framebuffer memory map. For rendering and copying operations the actual buffer addresses can be on any pixel boundary.

Double Buffering

For general real-time 3D graphics rendering with OpenGL, double buffering [3Dlabs95a] is needed to perform the real-time animation. The left of Figure 2 shows the framebuffer memory map for the highly configured example with a 1024x768 double buffered system with 32bit main planes (RGBA) in full screen.

Quad Buffering

Quad-buffered memory organization for the framebuffer is somewhat similar to the double-buffered system. There are, however, some key differences.

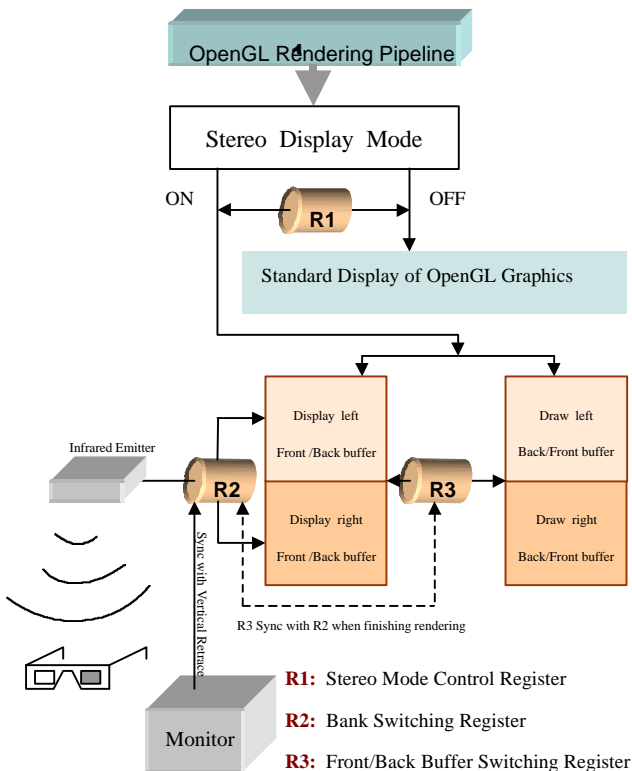


Figure 1 Conceptual Stereo Display HW Architecture

Figure 1 illustrates a flow diagram of an idealized version of stereo display pipeline of the system.

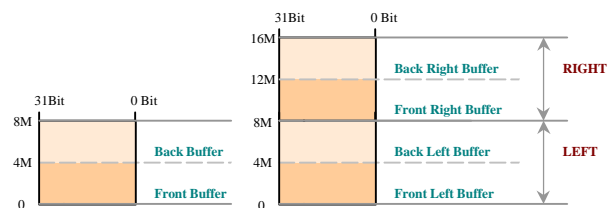


Figure 2 Double-buffered (left picture) and Quad-buffered (right picture) memory organizations

As was mentioned before, one stereoscopic graphics can be viewed if only both left image and right image, a stereo pair, are displayed simultaneously. Thus two double-buffered memories are required for stereoscopic display. That is, the framebuffer must be divided into four buffers which include front left buffer, back left buffer, front right buffer, back right buffer. Each individual double-buffered memory is employed to store the information of either left-eye image or right-eye image. This architecture of framebuffer organization is called quad buffering.

To display high-resolution stereo, the capacity of framebuffer for stereoscopic display needs to be twice as large as a monoscopic system. Taking the previous double buffering example, 8MB framebuffer is sufficient to display a common real-time graphics image with resolution 1024x768x32. However, such

capacity is only for one of a stereo pair. Under the same configuration, 16MB is needed for stereographics, as shown in Figure 2.

2.2 Stereoscopic Synchronization Control

There are two key synchronal controls to guarantee correct stereoscopic display:

◆ Hardware Synchronization Control

Left buffer and right buffer switching must synchronize with vertical retrace of the CRT.

◆ Software Synchronization Control

Simultaneously switch two pairs of double buffers after finishing rendering each image.

Hardware Synchronization Control

In order to display the correct stereoscopic images on a CRT, left eye can only view left image output from the left buffer; right eye can only view right image output from the right buffer. Each eye view should be refreshed often enough to avoid flickers. Time multiplexed stereo device such as liquid crystal stereo glasses is usually used for viewing the stereo image pairs. They can guarantee each eye to only see the corresponding view.

Therefore, hardware switching between left buffer and right buffer must synchronize with CRT refreshing. Different eye views are shown quickly and alternately on a CRT. Generally, each eye-viewed image is shown steadily and viewer's eyes feel comfortable when CRTs operate at 120Hz, 60Hz for each eye.

Looking at the below Figure 3, when the stereo sync signal is high, the left-eye view is being displayed on the CRT; when it is low, the right-eye view is being displayed. External stereo viewing equipment uses this signal to synchronize its time multiplexing [Hodges93] of the video into the viewer's left and right eyes, creating the illusion of a stereoscopic presentation of the graphics images.

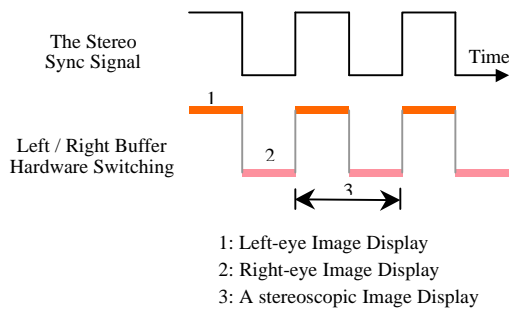


Figure 3 Synchronization control of the left buffer and the right buffer hardware switching

Software Synchronization Control

The switching between each of two double buffers is different from left and right buffer switching, which is fully controlled by hardware. Its synchronization control includes not only hardware control but also software control. Furthermore, software control is more important because it can control both software and hardware synchronization.

Strictly speaking, the time of the left scene rendering is not equal to that of the right in most cases. If the left scene has ended drawing first, the back left buffer and front left buffer will not be permitted to be switched until the rendering of the right scene has finished. Otherwise, the two images the viewer will see are not a stereo pair.

At the same time, switching the two double buffers after rendering must be operated synchronically while the left buffer and right buffer hardware switching. Otherwise, one of images is not entirely shown on the screen because of the possible unfinished scanning.

Both of the synchronization controls are performed through programming software, which is included in stereo graphics development kit and is completely transparent to the users. Thus, they are called Software Synchronization Control. Switching of

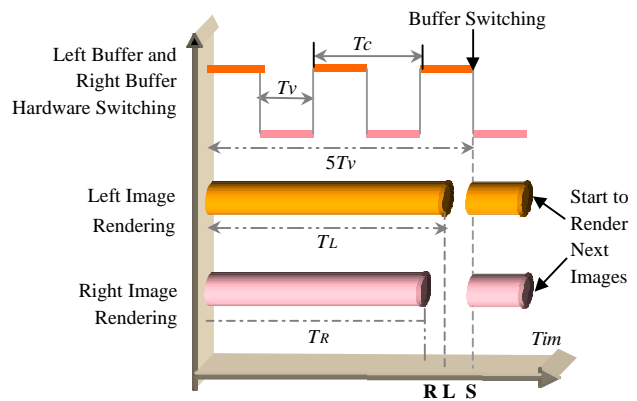


Figure 4 Synchronization control of two pairs of front buffer and back buffer switching

Front/Back quad pairs is allowed only after synchronization of Left/Right buffer switching.

From the example shown in Figure 4, it is obvious that it takes more time to render the left image than the right image, *i.e.* $T_L > T_R$, and $4T_v < T_L < 5T_v$. So their switching is operated in S time units, $5T_v$.

All in all, three pairs of buffers are performed to switch simultaneously between each pair by hardware control if only both image renderings are complete. These three pairs are left buffer and right buffer, back left buffer and front left buffer, and back right buffer and front right buffer respectively.

3. SOFTWARE DEVELOPMENT

3.1 Hardware Driver and SDK

Based on the stereo hardware architecture, a stereo hardware driver and graphic development kit is programmed for developing various stereo graphics applications. All of these applications can be run on a PC system with the stereo accelerator at 800x600, 1024x768 and 1280x1024.

Figure 5 shows the software infrastructure of the stereo display system. The SDK provides full access to the stereo functions of the stereo accelerator. It does not replace an existing graphics API. Rather, it works cooperatively with OpenGL to manage the rendering of the real-time 3D stereo scene with four buffers: GL_FRONT_LEFT, GL_FRONT_RIGHT, GL_BACK_LEFT, and GL_BACK_RIGHT [ARB93]. However, a double-buffered monoscopic mode only contains GL_FRONT and GL_BACK buffers in standard OpenGL for Windows NT [Segal98].

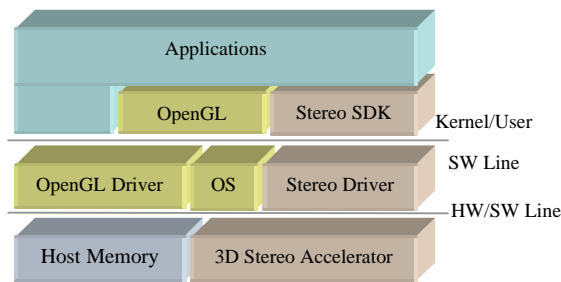


Figure 5 Software Infrastructure of the Stereo Display System

The kit includes stereoscopic mode control and display buffer control. Its main functions are:

- ◆ Testing whether the graphics hardware being used is stereo-ready
- ◆ Switching the system to stereo display mode
- ◆ Setting up full-screen stereoscopic graphics
- ◆ Restoring the system to non-stereo display mode
- ◆ Designating left or right buffer for rendering
- ◆ Swapping the stereoscopic double buffers
- ◆ Selecting any color buffer for rendering or display
- ◆ Switching positive and negative stereoscopic effect

It is easy for an application programmer to use the kit to directly control stereoscopic state and related devices. When the stereoscopic mode is selected, the graphics hardware automatically enables the shutter glasses and switches synchronously with the vertical retrace between the left and right front buffers.

When the stereoscopic mode is closed, the shutter glasses are deactivated and the display stops switching between left and right buffers.

The core parts are described in the following code in the library of the stereo SDK.

```
int agcStereoOn(void)
{
    //Test whether stereo driver is installed.
    if(!(agc3dntInit())) return 0;
    _READ_COLOR_BITS(PixelSize);
    _GET_VRAM_SIZE();
    // Calculate the each address of four buffers based on
    // the information of colorbits and the size of VRAM.
    _ASIGN_4_BUFFER_ADDRESS(Address);
    LeftBufferFlag = LEFT_FRONT;
    RightBufferFlag = RIGHT_FRONT;
    _STEREO_ON; // R1 register is set to ON
    _SYNC_SELECT_BANK0; // R2 register is set to BANK0
    return 1;
}

void agcStereoOff(void) //Exit to AGC_MultiBuffer Mode
{
    _DRAWFB_ADDR(0x0);
    _STEREO_OFF; // R1 register is set to OFF
    _SYNC_SELECT_BANK0; // R2 register is set to BANK0
    _SYNC_SELECT_FRONT; // R3 register is set to FRONT_BUFFER
    agc3dntExit();
}

void agcDrawBuffer(GLenum mode)
{
    if( mode == GL_LEFT)
        if(LeftBufferFlag)
            _DRAW_BUFFER(FrameBuffer[1]);
        else
            _DRAW_BUFFER(FrameBuffer[0]);
    else if( mode == GL_RIGHT)
        if(RightBufferFlag)
            _DRAW_BUFFER(FrameBuffer[3]);
        else
            _DRAW_BUFFER(FrameBuffer[2]);
    else return;
}

void agcSwapStereoBuffers(void)
{
    LeftBufferFlag =! LeftBufferFlag;
    RightBufferFlag =! RightBufferFlag;
    if(LeftBufferFlag) // R3 register is set to FRONT_BUFFER
        _SYNC_SELECT_FRONT;
    else // R3 register is set to BACK_BUFFER
        _SYNC_SELECT_BACK;
}
}
```

Stereoscopic display functions can also be easily added to an existing application or modified in an old application. Some external functions included in the kit are prepared for other special applications.

3.2 Programming with SDK

The stereographics programming consists of these major steps: (1) testing and enabling the stereoscopic display mode, (2) creating and managing a stereo pair of distinct left and right views, and (3) Disabling the

stereoscopic state when exiting or switching to normal display mode.

The first and last steps are very simple. Creating a stereo pair of views is different from creating a single view. Based on the theory of stereo vision [Carrozzo97, McAllister97], the stereo sense depends on horizontal parallax produced by the distance between left eye and right eye. Thus, programmers need to create parallel viewing pairs of perspective projection corresponding to left and right eye positions. The viewing pairs should be separately rendered in their corresponding buffers. `GL_LEFT` mode is selected as the parameter of `agcDrawBuffer()` to indicate that the left view is to be rendered only in the left buffer, and `GL_RIGHT` is only for the right buffer. When the renderings of both views are complete, `agcSwapStereoBuffers()` is called as normal double-buffer-swapped control in OpenGL for windows NT.

Here is an example for developing 3D stereo graphics using the stereo SDK.

```
agcStereoOn();
...
agcDrawBuffer(GL_LEFT);
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
gluLookAt(-dx,...);
DrawMyScene();
agcDrawBuffer(GL_RIGHT);
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
gluLookAt(dx,...);
DrawMyScene();
glFinish();
agcSwapBuffers();
...
agcStereoOff();
```

A stereo pairs of views are created with `DrawMyScene()` and `gluLookAt()` functions. `DrawMyScene()` is programmed for creating a single view. `gluLookAt()` is used for creating the parallax and its first parameter `dx` is half of the distance between the two eye points.

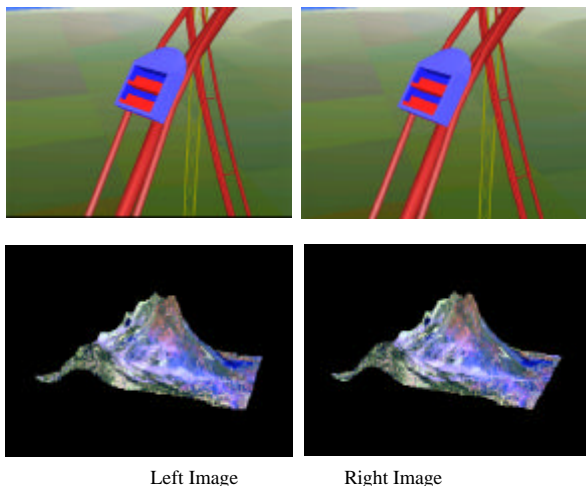


Figure 6 Two pairs of real-time 3D stereographics

There are two stereoscopic examples as as illustrated in Figure 6. The above stereo pair is an example modified from the roller coaster demonstration of by the 3Dlabs Inc. The sample below is directly programmed with OpenGL and the stereo SDK. Both of them can be controlled with mouse or keyboard in real-time with a stereo accelerator.

4. PC-BASED STEREOGRAPHICS SYSTEM

The stereo graphics system based on PC is a low-cost system. Many of its components are very common devices such as the CRT monitor and the host computer. It is usually made up of four parts:

- ◆ Stereo graphics accelerator
- ◆ Stereo graphics driver and development kit
- ◆ Stereoscopic viewing device
- ◆ Stereoscopic display device

Figure 7 shows the stereographics accelerator, shutter glasses and infrared emitter designed by our research group. This system is flexible and has some extended stereoscopic applications which will be mentioned later in section 4.4.

4.1 Stereographics Accelerator and SDK

The stereoscopic accelerator, with its driver and development kit, is the core of the stereographics system. Many real-time desktop VR applications can be developed with the stereographics kit combined with C++ and OpenGL under Windows NT.



Figure 7 Stereographics accelerator and stereoscopic viewing device

4.2 Stereoscopic Viewing Device

Stereoscopic images are output from graphic accelerator to CRT. Just as mentioned before, the stereo signal is synchronized with configured vertical scanning frequency of the CRT. The stereoscopic viewing device consists of an infrared emitter and a pair of infrared glasses, or a controller (integrated by the emitter) and a pair of LCD glasses with wires.

4.3 Stereoscopic Display Device

With the time-multiplexed stereo video, the CRT display device with high resolution and required refresh frequency must be provided for stereoscopic image display. CRT monitors and CRT projectors are



Figure 8 CRT monitor for stereoscopic display

used widely, as shown in Figure 8 and Figure 9. Projection-based displays can supply larger field of



Figure 9 CRT projector for stereoscopic display

view (FOV) and more distance from the viewer [Czernuszenko97].

4.4 Extended Stereoscopic Systems

Based on the design concept of stereographics accelerator, special extended stereoscopic systems can also be designed:

- ◆ Dual Stereographics Accelerators on One PC

The main purpose of this design is to widen the range of stereoscopic image display or to widen the stereoscopic FOV.

- ◆ One Stereographics Accelerator and One Non-Stereo Accelerator on One PC

It is employed for special applications such as geographic information system (GIS). The stereo images generated by the stereographics card are displayed on one monitor, and another monitor is used for displaying the common 2D or 3D graphics related to the process or the results of the operating stereoscopic images through a non-stereo accelerator.

- ◆ Stereographics System Composed of Dual Cards

and Dual PCs

In some cases, the performance of one PC limits the speed of the rendering of stereo images. A stereographics system with two sets of non-stereo accelerators and PCs can be designed to improve the rendering speed. One set for drawing the left view image, and the other for the right view. Again, the synchronization control is the key technology in this system, but is more complex than that of the single stereographics system.

5. CONCLUSIONS

This paper describes the architectures, features, and development methods of the real-time high-resolution stereo graphics system. Fully exploiting real-time stereoscopic 3D graphics display system based on PC is a significant challenge. It is essential to design stereoscopic display hardware architecture according to different demands in kinds of stereo graphics PC systems. These systems are required to be not only low-cost but also scalable, and can have widespread applications in VR, GIS and visual simulations. To summarize, these systems have the following major advantages:

High Resolution, Stable Stereoscopic Viewing

The left image and right image for stereoscopic viewing are rendered through graphic HyperPipeline in graphic processor, and double-buffered stereoscopic display is performed in framebuffer. For example, when the capacity of framebuffer is 32MB, the maximum resolution of double-buffered stereoscopic display is 1024x768x32bit, and the highest vertical rate of stereoscopic display is 120MHz, so the stereoscopic viewing is very stable.

Easy to Develop Application with the Stereo SDK

The stereo SDK is similar to the standard OpenGL functions. The functions are easy to use as a supplement to OpenGL for stereoscopic display support in Windows NT [ARB93][Segal98].

Real-time 3D Stereoscopic Viewing for Virtual Reality on PCs

With emitter and shutter glasses, this system is a low-cost 3D stereoscopic viewing system on PCs. 3D graphics can be controlled in real-time. This method is a key component in VR applications. Especially, it will play an important role on systems such as Immersive workbench [McDowall97], which are ideally suited to collaborative applications.

6. ACKNOWLEDGMENTS

We would like to thank Adward Zhang for contributing to part of the ideas, and Zejian Liu and Steve Deng for implementing hardware design of the

stereo accelerator. Special thanks go to Jame Tong, CEO of the Microstep Inc and Sunstep Inc, for providing the opportunities and funds for the development of these systems.

REFERENCE

- [3Dlabs95a] 3Dlabs Inc. *GLINT 500TX Hardware Reference Manual*, 3Dlabs Inc. San Jose, 1995
- [3Dlabs95b] 3Dlabs Inc. *GLINT 500TX Programmer's Reference Manual*, 3Dlabs Inc. San Jose, 1995
- [ARB93] OpenGL Architecture Review Board. *The OpenGL Reference Manual*, Addison-Wesley Press, Menlo Park, 1993.
- [Burdea94] Grigore Burdea, Philippe Coiffet. *Virtual Reality Technology*. John Wiley & Sons, Inc. USA, 1994
- [Carrozzo97] Mauro Carrozzo, Francesco Lacquaniti. Geometric Transformations for displaying virtual objects on stereoscopic devices. *Comput. & Graphics*, Vol.21, No.3, 1997, pp.324-334.
- [Czernuszenko97] Marek Czernuszenko, Dave Pape, etc. The ImmersaDesk and Infinity Wall Projection-Based Virtual Reality Displays, *Computer Graphics*, May 1997, pp.46-49.
- [Hodges93] Larry F Hodges. Tutorial: Time-Multiplexed Stereoscopic Computer Graphics. *IEEE Computer Graphics & Applications*, March 1993, pp.10-20.
- [McAllister97] David F. McAllister, Lou Harrison, Martin Dulberg. Stereo Computer Graphics for Virtual Reality, *SIGGRAPH 97 Conference Proceedings*, Course Notes 6, 1997
- [McDowall97] Ian E. McDowall, Mark T.bolas. New Developments for Virtual Model Displays, *Computer Graphics*, May 1997, pp.50-52.
- [Neider93] Jackie Neider, Tom Davis, Mason Woo. *OpenGL Programming Guide*, Addison-Wesley Press, Menlo Park, 1993.
- [Segal98] Mark Segal, Kurt Akeley. The OpenGL® Graphics System: A Specification (Version 1.2). Silicon Graphics, Inc. CA, March 1998.