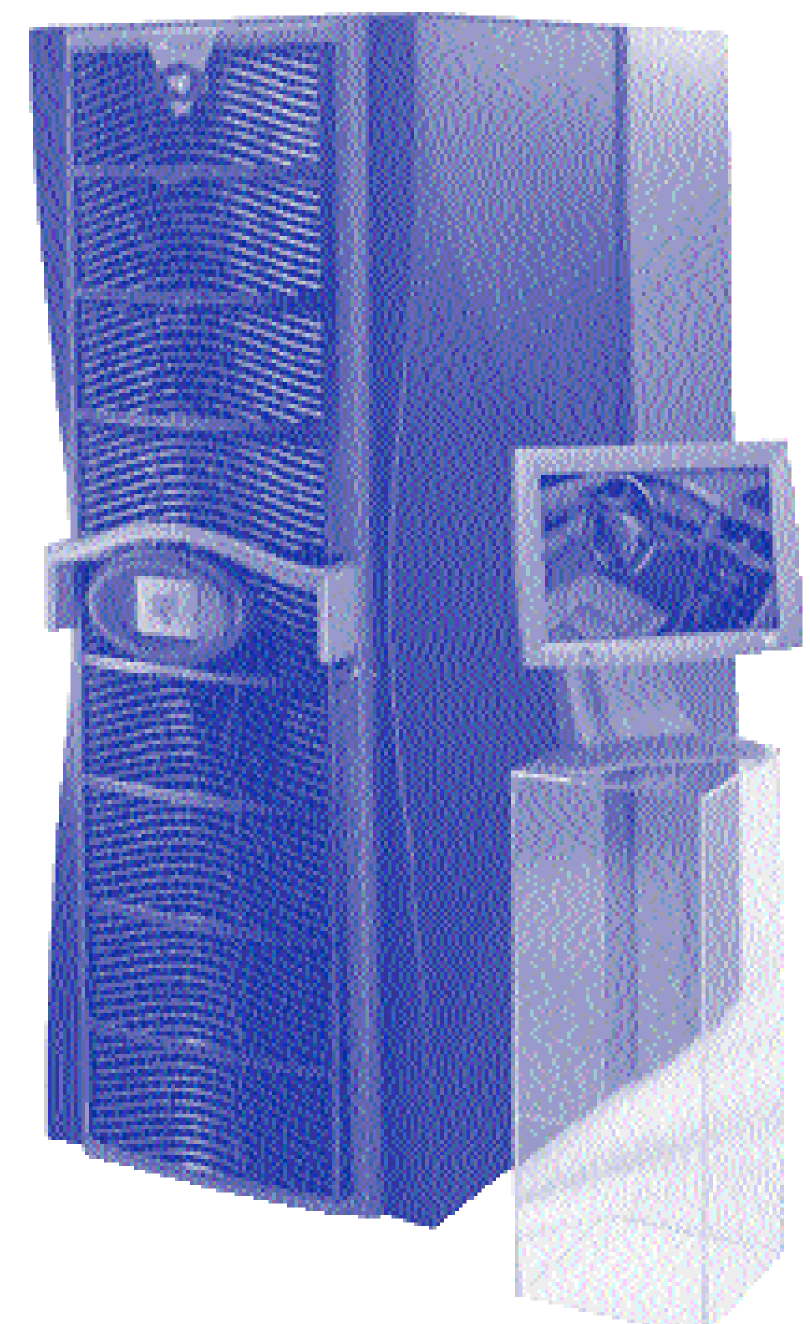
SGI™ Onyx® 3000 Series

Technical Report



SGI™ Onyx® 3000 Series

Technical Report, May 2001

Technical Report, May 2001

Corporate Office
1600 Amphitheatre Pkwy.
Mountain View, CA 94043
(650) 960-1980
www.sgi.com

North America 1(800) 800-7441
Latin America (52) 5267-1387
Europe (44) 118.925.75.00
Japan (81) 3.5488.1811
Asia Pacific (65) 771.0290

3087 (5/01) J11545

SGI™ Onyx® 3000 Series Technical Report

May 2001

TABLE OF CONTENTS

TABLE OF CONTENTS

(Intentionally left blank)

LIST OF FIGURES

(Intentionally left blank)

LIST OF TABLES

(Intentionally left blank)

1.0 Introduction

SGI, the recognized leader in developing sophisticated visual computing systems, is dedicated to advancing computer visualization technology now and in the future. After more than a decade, hardware and software solutions from SGI continue to define the state of the art in visual computing for creative professionals, scientists, and engineers. The SGI Onyx 3000 series is the latest offering in the most advanced line of binary-compatible visualization systems available. These systems establish a new high-water mark for their ability to process 2D, 3D and video data in real time, making them true visualization pipelines rather than simply graphics subsystems.

1.1 About This Document

This document serves as a detailed report on the SGI Onyx 3000 series. The SGI 3000 family, which includes the SGI Onyx 3000 series of visualization systems and the SGI™ Origin™ 3000 series of servers, implements the SGI™ NUMA architecture. For in-depth technical information on the CPU, memory, and I/O subsystems of the SGI Onyx 3000 series, see the SGI Origin 3000 Series Technical Report. For more information on virtual reality and SGI™ Reality Center™ facilities, see the SGI worldwide Web site (*www.sgi.com*).

1.2 InfiniteReality™ Graphics System

For the sake of simplicity, this document refers only to InfiniteReality3™, not the InfiniteReality or InfiniteReality2™ graphics systems. However, everything contained herein also applies to InfiniteReality and InfiniteReality2. Note that the InfiniteReality Geometry Engine® board is the GE14, and the InfiniteReality2 and InfiniteReality3 both use the GE16 Geometry Engine board. The InfiniteReality Raster Manager is the RM7; the InfiniteReality2 Raster Manager is the RM9; and the InfiniteReality3 Raster Manager is the RM10.

InfiniteReality3 is not a new architecture. InfiniteReality2 represents a performance enhancement over InfiniteReality, providing improved geometry, image processing, and anti-aliased pixel fill performance. InfiniteReality3 offers further performance enhancements and increased realism through increased texture memory (TRAM).

1.3 System Configurations

The SGI Onyx 3000 series is available in three different base configurations: SGI™ Onyx® 3200, SGI™ Onyx® 3400, and SGI™ Onyx® 3800. All of these products include the InfiniteReality3 graphics subsystem and take advantage of the unrivaled I/O bandwidth afforded by the SGI Onyx 3000 series host platform. The SGI Onyx 3000 series provides a scalable system architecture, with each increment of processing power adding up to 8GB additional memory, 650MB/second of memory bandwidth, 711MB/second of interprocessor communications bandwidth, and 711MB/second of I/O bandwidth. SGI Onyx 3000 series systems are available in the configurations given in Table 1.

Table 1. SGI Onyx 3000 Series Configurations

SGI Onyx 3000 Series Product	Type of R-Bricks Required	Number of CPUs	Form Factor	Number of Graphics Pipes
SGI Onyx 3200	None	2-8	Single Rack	1-2
SGI Onyx 3400	6-Port	4-32	1 CPU Rack; 1 I/O Rack; 1-4 Graphics Racks*	1-8
SGI Onyx 3800	8-Port; Meta (>128 processors)	16-512	Multirack; up to 8 Graphics Racks*	1-16

*Up to two G-Bricks in a single graphics rack

A system must have at least one C-brick and either one I-brick or one X-brick for each graphics pipe, not each G-brick. Each G-brick may contain two graphics pipes; one graphics pipe with one or two Raster Managers (RMs) and one graphics pipe with one, two, or four RMs.

1.3.1 SGI Onyx 3200

SGI Onyx 3200 provides a single-rack configuration (see Figure 1) that is deployed predominantly as a departmental resource to serve a wide range of visual and numeric computing needs. Because of the modular, highly scalable architecture, SGI Onyx 3200 can grow to meet an organization's diverse and ever-expanding needs. This system includes two to eight 8MB cache MIPS® R12000® processors; 512MB to 16GB of memory; one graphics pipeline with one, two, or four 80MB RMs providing 256MB of texture memory and up to 320MB of frame buffer capacity; and one graphics pipeline with one or two RMs providing 256MB of texture memory and up to 160MB of frame buffer capacity, and a superwide, ultrahigh resolution monitor supporting up to 1920x1200.

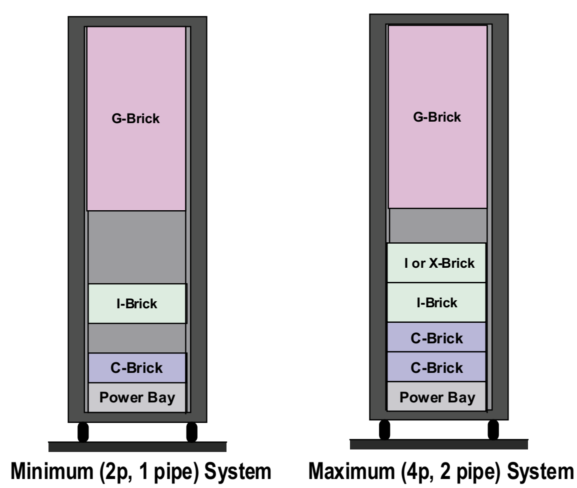


Figure 1. SGI Onyx 3200 Configurations

1.3.2 SGI Onyx 3400

SGI Onyx 3400 is a flexible system that can handle many visualization tasks and has the potential to grow into a more powerful system as increasingly complex challenges are addressed. SGI Onyx 3400 includes four to 32 MIPS R12000 processors, up to eight InfiniteReality3 visualization pipelines, and 64GB of main memory. Although this system can start as a single rack, it can grow to a six-rack system as processors, I/O and graphics pipes are added (see Figure 2). SGI Onyx 3400 is ideal for a wide range of multipipe visualization applications.

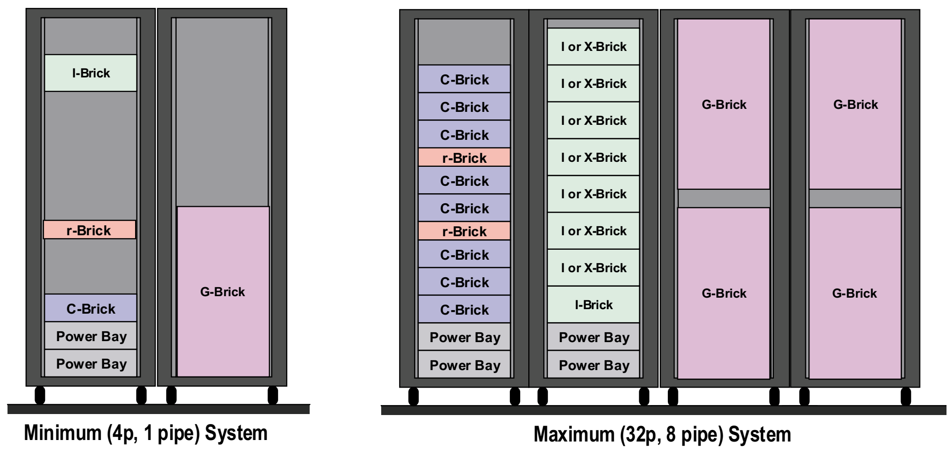


Figure 2. SGI Onyx 3400 Configurations

1.3.3 SGI Onyx 3800

SGI Onyx 3800 is a truly scalable workhorse for addressing all visual supercomputing needs (see Figure 3). This system includes from 16 to 512 MIPS R12000 processors, up to 16 InfiniteReality3 graphics pipes, and 1TB of main memory, all seamlessly interconnected to form a single system image. SGI Onyx 3800 is the ultimate answer to the toughest problems.

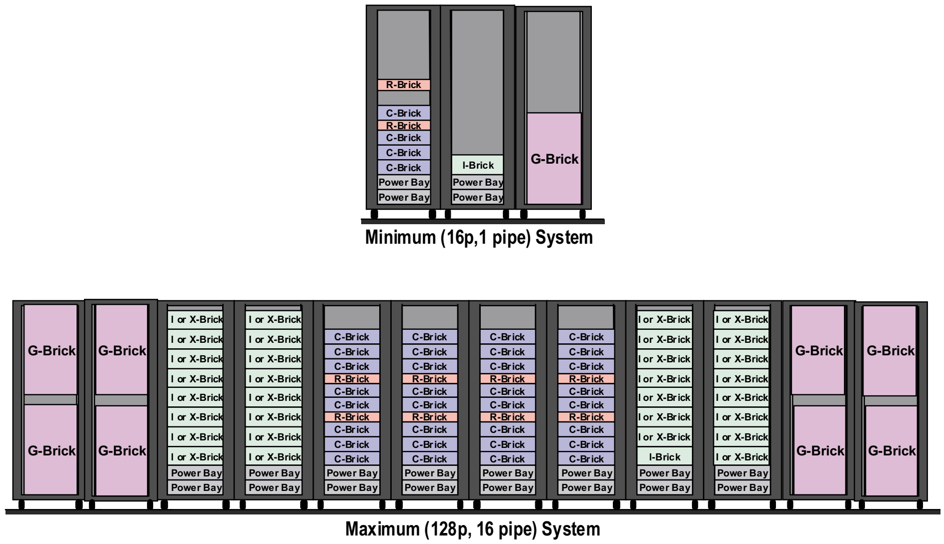


Figure 3. SGI Onyx 3800 Configurations

The SGI Onyx 3000 series was designed specifically to deliver flexibility and extensibility, so a single multipipe system may be deployed in different operating modes to solve different problems. As described below, these modes include supercomputer, multiseat, multidisplay, and multipipe rendering. In most cases, switching from one mode to another is accomplished exclusively in software; no hardware reconfiguration is required. OpenGL Vizserver™ further enhances this flexibility by breaking down the physical barriers of hard-wired monitors and users at a distance from the host that in the past may have limited access to the power of the SGI Onyx 3000 series. This software acts as an application-transparent interface between the InfiniteReality graphics and SGI as well as non-SGI desktop clients distributed across commodity networks. Through OpenGL Vizserver, access to the SGI Onyx 3000 series can be delivered anywhere on the organization's network.

1.3.4 SGI Onyx 3000 Series and Multipipe Rendering Mode

SGI Onyx 3000 series systems with InfiniteReality3 graphics can be tasked to focus all pipelines on a single rendering window, resulting in near-to-perfect linear scalability of visualization performance (where visualization performance includes geometry rate, pixel fill rate, texture capacity, and/or texture paging bandwidth). This application of multiprocessing principles to visualization can be accomplished with the SGI Onyx 3000 series in two ways: with the DPLEX (Digital Multiplexing) hardware option or with MonsterMode software.

DPLEX is an optional daughtercard for SGI Onyx 3000 series systems that enables digital multiplexing of two or more InfiniteReality pipelines. One DPLEX option is required for each pipeline in the system. The concept is simple: multiple pipelines operate simultaneously on successive frames of an application, which are then digitally multiplexed together before being converted to analog video. Key applications for DPLEX include distortion correction (required for dome simulators) and interactive large model visualization. For suitable applications, an n-pipe system equipped with DPLEX can support n times the frame rate of a single-pipe system running the same application.

MonsterMode is the collective name of several software-based methods for distributing a data set over multiple pipelines. Each uses the high-bandwidth memory subsystem of the SGI Onyx 3000 series rather than special hardware (DPLEX) for the interpipeline transfers required to partition the data and compose the final image. MonsterMode rendering methods include 2D composition for handling polygonal models and 3D composition for handling volumetric models. 3D composition provides the further benefit of additive texture-mapping resources: an n-pipe system using MonsterMode 3D-composition software has n times the effective texture memory and texture download bandwidth of a single-pipe system (up to 4GB and 5GB/second, respectively, for a 16-pipe configuration).

Applications will require modification to support DPLEX or MonsterMode operation. Full-screen applications written to OpenGL Performer™ or OpenGL Optimizer™ 1.1 will require the least work. For details, please consult your SGI representative.

1.3.5 SGI Onyx 3000 Series in Multiseat Mode

With the addition of extra keyboards and mice (and USB hubs and extenders if required), multipipe SGI Onyx 3000 series systems can be deployed in multiseat mode. In this mode, multiple simultaneous users each have exclusive control over an InfiniteReality3 graphics pipeline as well as shared access to all other system resources, including memory, disk, and compute. Multiseat mode provides a solution for users who require more memory capacity, disk storage, graphics performance, or processing capabilities than are afforded by desktop workstations.

1.3.6 SGI Onyx 3000 Series in Multidisplay Mode

Each InfiniteReality3 graphics pipeline ships standard with a 2-channel display generator and may be ordered with an optional 8-channel display generator, so every SGI Onyx 3000 series configuration is capable of driving multiple displays. Additionally, a suite of tools permits users to choose from a wide range of video formats and format combinations to suit any application or display technology. The SGI Onyx 3000 series is therefore ideal for driving virtual reality devices such as SGI Reality Center facilities, CAVEs, visual simulators, Power Walls, head-mounted displays, and other advanced interfaces in both monoscopic and stereoscopic formats.

2.0 SGI Onyx 3000 Series with InfiniteReality3 Graphics Overview

The SGI Onyx 3000 series is designed to concurrently process 3D geometry, imagery, and video data at real-time rates, making it a series of true visualization pipelines. The architecture underlying the SGI Onyx 3000 series combines raw processing power, an advanced feature set, and record-setting throughput for unrivaled visualization performance. Differentiated capabilities include:

1. Hardware-accelerated image processing
2. High-quality, multisampled, full-scene, anti-aliased pixel fill capacity
3. Hardware tuning for complex geometry
4. Real-time texture paging from main memory or disk
5. Scalable visualization capabilities using multipipe rendering methods for higher geometry rate, pixel fill rate, texture capacity, and texture paging bandwidth
6. Flexible display subsystem
7. Tightly coupled, host-integrated computer image generator (HI-CIG) architecture

The InfiniteReality3 graphics subsystem comprises three unique boards: Geometry Engine, the Raster Manager (RM), and the Display Generator (DG). Geometry Engine provides the interface to the compute subsystem and is responsible for the preliminary geometric transformations (translation, rotation, scale) and lighting calculations for three-dimensional data. It also performs image processing functions (convolution, histogram equalization, orthorectification, and more) on two-dimensional images. The RM takes the results from Geometry Engine and scan-converts the data into a digital image. The RM performs various pixel operations, including Z-buffer testing, stencil testing, color and transparency blending, texture mapping, and multisampled anti-aliasing. The RM has both texture memory for storing textures (images) and frame buffer memory for performing and storing pixel operations. Finally, the DG takes the output from the RM and converts the digital data into an analog signal for driving a display. A DG can be programmed to display different portions of the RM frame buffer to its two display channels. One of these display channels can be converted into a composite video mode and connected directly to a video tape recorder. Optional configurations of the DG support:

1. Up to eight analog output display channels (DG5-8)
2. Graphics to CCIR601 Serial Digital Video out (GVO)
3. Graphics to High-Definition (HD) and Standard-Definition (SD) Serial Digital Video Out
4. Low Voltage Differential Signal (LVDS) digital video out (DPLEX)

.

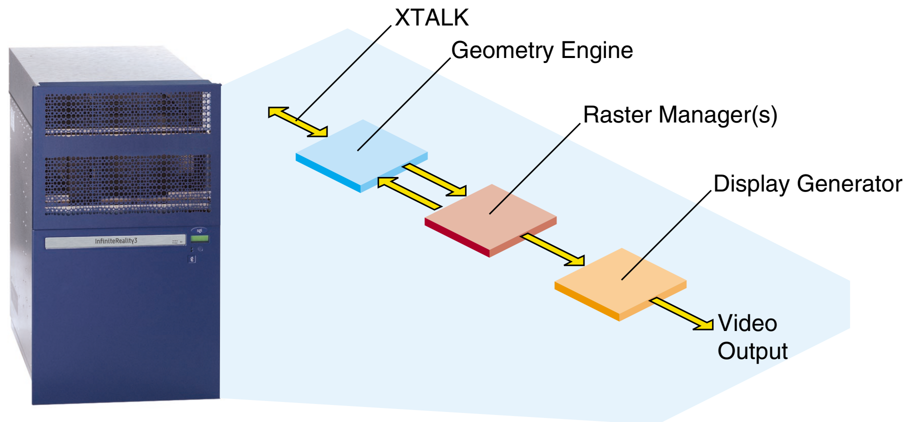


Figure 4. SGI Onyx 3000 Series Visualization Pipeline

3.0 Graphics Technologies

3.1 OpenGL®

The native graphics programming interface for the SGI Onyx 3000 series is OpenGL. OpenGL is the premier environment for developing high-performance 2D and 3D graphics, imaging, and video applications. The OpenGL application programming interface (API) is a vendor-neutral, multiplatform industry standard that provides access to a rich set of graphics capabilities, including transformations, lighting, atmospheric effects, texturing, anti-aliasing, hidden surface removal, blending, and pixel processing, among many others. OpenGL is designed to be independent of operating systems and window systems and is supported on virtually all workstations and personal computers available in the industry today. To ensure that OpenGL remains the industry standard, SGI has open sourced the OpenGL Sample Implementation to make it even easier to use together with X11 and XFree86 as well as other environments that may emerge in the future. The OpenGL Sample Implementation is both a reference implementation and a driver framework used by almost all commercial 3D hardware vendors to develop hardware drivers for their systems.

Among OpenGL's many desirable characteristics is its extensibility; OpenGL capabilities have been extended to include dozens of special functions for applications in visual simulation, imaging, volume rendering, advanced industrial design, film and video effects, and animation. In addition, OpenGL provides the foundation for higher level programming layers. OpenGL Performer, OpenGL Optimizer, and OpenGL Shader™, for example, are supported by OpenGL.

3.2 OpenGL Performer

OpenGL Performer is a high-performance, portable 3D rendering toolkit for developers of real-time, multiprocessed, interactive graphics applications for the range of SGI graphics systems. OpenGL Performer simplifies development of complex applications for visual simulation, simulation-based design, virtual reality, interactive entertainment, broadcast video, CAD, and architectural walk-through. OpenGL Performer guarantees the highest performance with automatic, optimal use of available components and features of all IRIX® and Linux® OS-based SGI machines. This guarantee includes the use of multiple CPUs, multiple graphics pipelines, fast filesystem and disk access, and real-time scheduling.

Key features include:

- Real-time fixed frame rate operation for truly immersive simulations
- Automatic scene and load management features: view frustum culling, Level-of-Detail (LOD) evaluation, and Dynamic Video Resolution (DVR)
- Run-time and real-time profiling for use in debugging, tuning, and load management
- Asynchronous paging of database geometry and imagery, including fast loading formats
- Database interoperability with run-time linking to database loaders for loading data files of any or multiple file format(s)
- Dynamic Texture Resolution (DTR) for automatic paging and management of huge textures, up to 8Mx8M texels in a single texture (Cliptexturing)

- Active Surface Definition (ASD) for automatic paging and evaluation of terrain with continuous LOD evaluation
- Asynchronous database intersection and user database processing
- Lightpoints and calligraphic points for visual simulation
- Atmospheric effects for visual simulation, including fog, haze, and support for implementing range/angle correct layered fog
- Support for special effects for realistic and dynamic environments
- Support for programmable shading
- Video textures
- Cross-platform support

To help developers get the most out of OpenGL Performer, sample applications and viewers are included with source code, and more than a gigabyte of utility libraries, file loaders, examples, and data, much contributed by third-party suppliers, is included with OpenGL Performer.

3.3 OpenGL Optimizer

OpenGL Optimizer is a programmer’s toolkit that provides the key enabling software technologies required for creating and interacting with extremely large and complex 3D data sets. This toolkit includes an API that offers access to a rich library of advanced graphics algorithms and a suite of tools that make it easier for application developers to create CAD/CAM, digital content creation, and digital prototype applications.

Additionally, OpenGL Optimizer provides high-performance rendering and interaction with large models while maintaining the high model fidelity demanded by CAD and other engineering, scientific, and technical applications. Providing attention to detail while maintaining high performance in a cross-platform environment is the primary strength of OpenGL Optimizer. OpenGL Optimizer is a comprehensive and profoundly innovative API for large model visualization and interaction that sets new standards for computer-aided design.

Key features include:

- Scene graph culling: object removal technologies such as view frustum, occlusion, and back patch culling
- Higher order representations: Coons patches, Hermite cubic spline patches, NURBS patches, spheres, tori, and many others
- Topology representation: topological relationships between surface patches, such as shared edges and junctions (used to provide crack-free tessellation at shared boundaries of surface)
- Integrated multiprocessing: high-performance rendering harnessing multiple CPUs
- Integration with scene graph API: OpenGL Optimizer is built on the industry-standard OpenGL API and a scene graph layer API; it is designed to allow the programmer to use both concurrently
- High-quality rendering: unique advanced shading and reflection mapping capabilities

- Polygonal simplification: new advanced simplification technology known as the Successive Relaxation Algorithm allows developers control over high-quality polygon mesh reduction and simplification
- Surface tessellation: fast and accurate crack-free tessellation delivers both low polygon counts and high quality

3.4 OpenGL Shader

OpenGL Shader enables interactive programmable shading for increased realism using hardware acceleration on a standard OpenGL graphics card. OpenGL Shader treats the OpenGL architecture as a general single instruction, multiple detail (SIMD) computer and translates the high-level shading description into OpenGL rendering passes. Once a high-quality rendering effect has been developed, improved performance and quality are automatically achieved as the underlying hardware evolves.

A special-purpose language is used to construct programs that describe a rendering effect. All aspects of the object and its related environment may be encapsulated within these programs, referred to as shaders. Examples of relevant aspects include the color; material properties, including reflectivity; the light source position and its emission characteristics; and the transmittance properties of the atmospheric media. The combination of these programs allows each point on an object to be rendered with the resulting color, or opacity, as seen from a specified viewpoint.

OpenGL Shader introduces a compiler between the application and the graphics library. The compiler can simply produce a general set of rendering passes; however, knowledge of the target hardware can be exploited by the compiler to pick an optimized set of passes. The OpenGL Shader approach treats the graphics hardware as a collection of fundamental operations that can be combined to yield a final result in multiple passes. This high-level programming of complex effects needs to be performed only once, yet this approach allows a single set of complex effects to be easily retargeted to different rendering hardware.

Key features include:

- Real-time performance supports interactive shader design and application
- Compact library can easily be invoked from any application
- Simple interface results in minimal invasiveness
- Output of OpenGL source code can be compiled directly into an application
- Example applications and sample shaders included

3.5 OpenGL Vizserver

OpenGL Vizserver is a technical computing solution designed to deliver advanced visualization capabilities and performance to the desktop. Implementing an SGI Onyx 3000 series system as a graphics server, OpenGL Vizserver allows users to view and interact with large data sets from a desktop system at any location in an organization. OpenGL Vizserver is designed primarily for users involved in scientific data visualization, including seismic data analysis, and in manufacturing design and engineering. OpenGL Vizserver enables a single

SGI Onyx 3000 series system to distribute visualization sessions to multiple UNIX® operating system desktop workstations. With OpenGL Vizserver, graphics processing is handled entirely on the SGI Onyx 3000 series system.

Key features include:

- Application transparency: well-coded OpenGL applications work without change
- Exploitation of the high-end, high-quality capabilities of the InfiniteReality3 graphics
- High performance: up to 17 frames/second at 1280x1024 OpenGL window size over Gigabit Ethernet
- Leverage of leadership compute, I/O, and system architecture of the SGI Origin 3000 series servers
- Tuned for general-purpose switched 100Base-T, ATM, and Gigabit Ethernet networks
- Application independence across all supported clients
- Packet-filtering firewall traversal
- Dual head client support
- Unlimited client licenses for desktops running IRIX 6.5, Sun® Solaris™ 2.6 and Red Hat® Linux 6.2, with support for Microsoft® Windows NT® 4.0 expected in summer 2001

The two main components of the OpenGL Vizserver architecture are the redirection of the OpenGL rendering to the host machines graphics pipes and the transport of the rendering results to the client. Under normal circumstances, the application would interface with the client via the Xserver. With OpenGL Vizserver, the application interfaces with the client Xserver through an OpenGL Vizserver layer, which redirects all OpenGL calls to the host machine's graphics subsystem. When a frame is done, indicated by either glXSwapBuffers() or glFinish(), the final image is pulled out of the pipe through the OpenGL Vizserver layer, compressed, and finally sent to the client. The OpenGL Vizserver client then receives and decompresses the frame buffer images from the server and displays them in the correct window.

3.6 OpenGL Volumizer™

OpenGL Volumizer is the industry's first commercially available high-level immediate mode volume rendering API for the energy, medical, and science markets. OpenGL Volumizer harnesses the SGI Onyx 3000 series to provide a powerful combination of features and capabilities previously unavailable to the industry. OpenGL Volumizer provides unprecedented features such as the ability to roam through extremely large data sets of up to 100GB. It also leverages available 3D texture-mapping hardware to boost application performance upwards of 10 to 100 times faster than existing CPU-bound solutions.

OpenGL Volumizer is built upon the concept of the voxel, a three-dimensional texel, as the fundamental small-scale volumetric primitive and the tetrahedron, a four-sided, three-dimensional solid object, as the fundamental large-scale volumetric primitive.

Key features include:

- Mixing volumes and surfaces with the same scene
- Arbitrarily shaped regions of interest

- Roaming very large data sets
- Unified approach to treatment of regular grids and unstructured meshes
- Ease of use and performance as a layer on top of OpenGL
- Highly flexible design with low-level services and high-level utilities
- Ability to easily use, configure, modify, and extend for application differentiation
- Support for the combination of volume visualization with opaque geometry
- Support for volume deformation, co-resident volumes, and time-varying volumes
- Compatibility with existing APIs such as Open Inventor™ and OpenGL Performer

3.7 ImageVision Library®

ImageVision Library is an object-oriented, extensible toolkit for creating, processing, and displaying images on all IRIX OS-based SGI visual workstations. ImageVision Library provides image processing application developers with a complete, robust framework for managing and manipulating images.

Today's rapidly advancing hardware technology often requires software developers to rewrite their code for each new hardware release. ImageVision Library is explicitly designed to provide a constant software interface to changing hardware. Applications written today will run on all future generations of machines, requiring few or no changes. With ImageVision Library, SGI continues to provide the benefits of next-generation hardware while maintaining all the advantages of a stable but growing software environment.

Key features include:

- Ability to read standard file formats, such as TIFF, PNG, GIF, JFIF, PhotoCD, PPM, Alias, SOFTIMAGE, and SGI formats
- Object-oriented and extensible
- Image Format Library (IFL), available for the IRIX and Windows® operating systems
- Image Operators (over 70 provided with library)
- Operators are interconnected to express sequences of operations
- Demand-pull data flow model
- Configurable caching for efficient handling of very large images
- Transparent acceleration using OpenGL and multiple processors
- Suitable for both batch and interactive processing
- Electronic Light Table (ELT) specialized support

3.8 dmSDK

Digital Media Software Development Kit (dmSDK) is a high-performance, cross-platform API for developers who want to take advantage of the digital media capabilities of SGI visual workstations. Digital media capabilities include high-definition and standard-definition video, audio, and IEEE 1394 interface and control.

dmSDK contains the development environment for the digital media libraries and provides information about the relevant data formats, how to program the audio and video I/O paths, compression/decompression, and audio/video synchronization. In addition, dmSDK provides for integration with OpenGL graphics capabilities to combine video and graphics for real-time video effects.

Key features include:

- Uniform programming interface for various media input and output devices
- Precise synchronization of video, audio, and graphics streams
- Improved portability across multiple operating systems
- Simplified compositing of 2D, 3D, and video information

The included demos, sample code examples, and documentation describe fully how to get the most out of dmSDK 2.0.

4.0 Geometry Subsystem

The first stage in the visualization system of an SGI Onyx 3000 series system is the geometry subsystem. It resides on the Geometry Engine board. The primary functions of the geometry subsystem are handling data transfer to and from the host, performing OpenGL command parsing, and executing geometry and pixel processing commands. The geometry subsystem provides dedicated hardware acceleration for display list processing. The geometry subsystem of the SGI Onyx 3000 series was designed to support the OpenGL API. Large FIFO buffers are provided to ensure that the geometry subsystem can continue executing without stalling, regardless of the state of other stages within the pipeline.

4.1 Geometry Engine

Geometry Engine processors are custom-designed by SGI. Two of four Geometry Engine processors are employed in multiple instruction, multiple data (MIMD) fashion within the geometry subsystem depending on the graphics option chosen. Each Geometry Engine processor contains three separate floating-point cores that execute in single instruction, multiple data (SIMD) fashion.

Geometry Engine processors do both geometry (per vertex) and pixel processing. Geometry processing includes vertex transformations, lighting, clipping, and projection to screen space. Pixel processing includes many common image processing operators, such as convolution, histograms, scale and bias, and lookup tables. Pixel operations can be applied to standard pixels, textures, or video.

(Intentionally left blank)

5.0 Raster Subsystem

The raster subsystem, residing on the RM boards, has most of the graphics systems custom VLSI processors and memory. Triangle, point, and line data received by the RM from the geometry subsystem must be scan-converted into pixel data, then processed into the frame buffer before a finished rendering is handed to the display subsystem for generation of displayable video signals.

By using extensive parallelism in most stages, the raster subsystem performs anti-aliasing, texture mapping, and image processing functions without encountering bottlenecks. Besides basic anti-aliasing for graphics primitives, the graphics system also supports full-screen anti-aliasing with a method called multisampling. With multisampling, images are rendered into a frame buffer with a higher effective resolution than that of the displayable frame buffer; the number of subsamples computed for each pixel determines how much higher. For example, if each pixel has eight subsamples, positioned at eight of the 64 possible subpixel locations, then the effective resolution is eight times the displayable resolution.

When the image has been rendered into these subsamples, all samples that belong to each pixel are blended together to determine pixel colors. Image memory is interleaved among parallel processors so that adjacent pixels are always handled by different processors. Thus, each polygon can have multiple processors operating on it in parallel.

5.1 Texture Processing

For each pixel, scan conversion produces texture coordinate data (S, T, R, W) that are sent to a dedicated texture processing unit. This hardware performs perspective correction on texture coordinates and determines the detail level for MIPmapping, producing addresses for the relevant area of texture memory. Texture data (texels) are stored in one of several formats, depending on the number and precision of color and transparency components desired. Texels in the area covered by a pixel are read and used in one of several user-definable texture filtering calculations. The resulting color value is sent to pixel processors along with the pre-texture color and depth values produced by the main scan converter. Texture color lookup tables are also supported, enabling hardware support for applications involving dynamic transfer functions such as sensor simulation.

5.2 Pixel Processing

At the next stage of rendering, texture color is used to modify the pre-texture primitive color according to a texture environment equation. Here is where texture can be used to decal, modulate, or blend with the underlying primitive color. The amount of fog at the current pixel is calculated from the depth value, and this fog color is blended with the textured color to produce the final color and transparency.

Pixel processors support 10-bit RGB and 12-bit RGBA components for pixels, providing a selection of more than 68 billion colors for any given object or vertex. 16-bit luminance-alpha or color index formats are also supported. This accurate color information is critical for image processing applications and is supported through ImageVision Library, as well as OpenGL.

The raster subsystem supports one to four RMs. More RMs increase the scale of both pixel fill performance and frame buffer resolution. (Texture memory remains constant since each RM uses its own texture memory rather than sharing it across RMs.) With one RM installed, SGI Onyx 3000 series systems with InfiniteReality3 graphics support the ultrahigh-resolution standard of 1920x1200 pixels noninterlaced and all HDTV formats. The ultrahigh-resolution 1920x1200 monitor is included on SGI Onyx 3000 series systems.

6.0 Display Subsystem

The video display subsystem for the SGI Onyx 3000 series takes rendered images from the raster subsystem digital frame buffer and processes pixels through digital-to-analog converters (DACs) to generate an analog pixel stream suitable for display on a high-resolution RGB video monitor. The display subsystem supports programmable pixel timings to allow the system to drive displays with a wide variety of resolutions, refresh rates, and interlace/noninterlace characteristics.

The standard configuration provides two independent video channels. Eight independent video channels are available as an option. These additional video channels can serve as standard RGB video channels. The second video channel can also serve as a composite video or S-Video encoder. Flexible built-in scan conversion allows the video encoder to process any rectangular area of any video channel, up to and including the full screen. Convenient letter-boxing lets the composite encoder work undistorted on video formats with aspect ratios other than the 3:4 aspect ratio of NTSC or PAL.

Advanced hardware in the video display subsystem performs real-time video resampling, giving powerful new methods of guaranteeing scene update rates and using frame buffer memory efficiently, as well as allowing full-screen recording to standard VCRs. Compatibility with a wide variety of video equipment such as projectors and recorders is provided by composite sync, sync-on-green, and separate horizontal and vertical sync signals.

Composite video and S-Video encoder outputs are for industrial or monitoring purposes only and are not intended for broadcast television use. For broadcast-quality composite video, a video channel may be configured as RS-170 or Euro 625 resolution and sent through an external, broadcast-quality encoder. When standard-definition or high-definition broadcast-quality component digital video is desired from the display subsystem, the Graphics-to-Video Out (GVO, described in Section 7.18.3) option can provide SD digital video output and the High-Definition Graphics-to-Video Out option (HD GVO, described in Section 7.18.5) can provide SD or HD digital video output.

The display system provides video support for the Japanese HDTV standard (1920x1080 pixels), both interlaced and noninterlaced, at 60 Hz. HDTV trilevel sync is provided for easy recording of HDTV signals without external signal conditioning equipment. Additionally, a special 72 Hz noninterlaced version of this format is provided for flicker-free viewing in ambient office lighting environments. The 1920x1080 pixel format is supported on the entry-level, 1-RM configuration of the SGI Onyx 3000 series. This video flexibility is supported by the standard system monitor, which is a high-resolution multisync display capable of displaying any video signal from VGA (640x480) up to the 1920x1200 format.

Video features of the SGI Onyx 3000 series with InfiniteReality3 graphics are easy to configure, using the Video Format Compiler and Video Format Combiner. See Section 7.14.4, Video Format Combiner, and Section 7.14.11, Programmable Video Timing, for more information.

6.1 ACCESS.bus

Typically, monitors have knobs that are used to adjust brightness, contrast, picture size, position, and color balance. The SGI Onyx 3000 series uses ACCESS.bus to provide this functionality via a program on the CPU or in an X Window interface. ACCESS.bus can also be used to read operational attributes from the monitor.

7.0 Graphics Features and Capabilities

7.1 Resolution

The SGI Onyx 3000 series supports a wide range of display resolutions. The base system includes one RM and provides 2.62 million pixels/second fill rate. 2-RM and 4-RM systems offer fill rates of 5.24 million and 10.48 million pixels/second, respectively. For example, a 1-RM system can drive a 1920x1200 display or two 1280x1024 displays. Alternately, a 2-RM system can drive six 960x720 displays or one 1280x1024 and six 800x600 displays. 4-RM systems maintain the display capabilities of the 2-RM system while increasing pixel quality and fill performance.

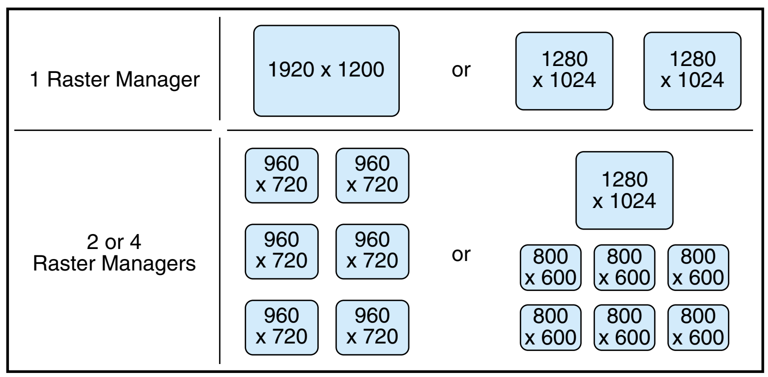


Figure 5. Display Resolution Examples for the SGI Onyx 3000 Series with InfiniteReality3 Graphics

The minimum amount of frame buffer memory you can allocate per pixel is 256 bits, which supports quad-buffered stereo with up to 12-bit RGBA and a 23-bit depth buffer, or 4-sample anti-aliasing at 10-bit RGB with a 23-bit precision depth buffer. Per-pixel memory can be doubled or quadrupled, with a corresponding reduction to one-half or one-quarter of available pixels.

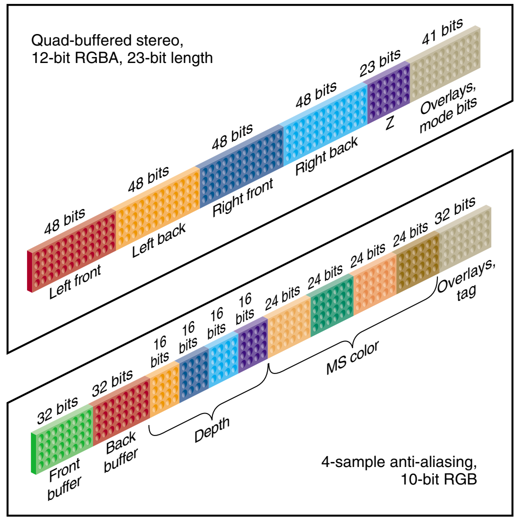


Figure 6. Choices of Allocation of 256 Bits of Frame Buffer Memory

7.2 Graphics Primitives

The general-purpose InfiniteReality3 graphics pipeline supports a wide range of graphics primitives for constructing and rendering real-time 3D objects. The basic primitives are polygons, vectors, points, and parametric polynomial surfaces (such as NURBS and Bezier patches). These primitives may be rendered with color, transparency, lighting, texture, and various surface properties.

For complex, contiguous surfaces such as terrain, triangles, or quadrilaterals may be combined into meshed polygon strips with common vertices. This reduces system transformation demands and efficiently uses the inherent parallelism in the InfiniteReality3 graphics pipeline.

.

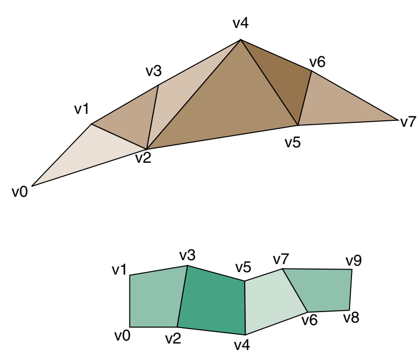


Figure 7. Meshed Polygon Strips

7.3 Immediate Mode and Display List Mode

OpenGL supports two mechanisms for delivering data to the rendering pipeline: immediate mode and display list mode. In immediate mode, transformations, geometry, and other data are delivered to the rendering pipeline directly from the application by calling a series of OpenGL routines. For example:

```
glBegin(GL_TRIANGLE_STRIP);
glVertex3fv(vertex)
...
glEnd()
```

Immediate mode has maximum flexibility because the application can recompute data when sending it to the rendering pipeline.

Additionally, the application can compile immediate mode commands into display lists, stored as part of the OpenGL rendering state. The application can execute display lists wherever it normally uses immediate mode commands. Rendering data is stored in an efficient representation and can be issued to the rendering pipeline with maximum performance. However, extra storage is required for the rendering data and some small amount of overhead is incurred in executing the list, so lists are not suitable for small amounts of data (a single triangle, for example). Finally, display lists are usually inappropriate for dynamic data because they need to be recompiled whenever data changes. While the basic character of display modes and immediate modes is the same across the SGI product line, some features and characteristics specific to the SGI Onyx 3000 series are worth noting.

7.3.1 Rendering Performance and Vertex Arrays

Rendering on SGI platforms is typically split into a three-stage pipeline:

- Data delivery: host to rendering pipeline
- Geometric processing
- Rasterization

Performance bottlenecks can occur in any of the three stages; however, performance may be improved by tuning the algorithms through which the application traverses its data and by using suitable OpenGL commands for delivering data to the pipeline. For example, rather than using floating-point data types for colors, normals, and texture coordinates, it may be more efficient to use short or byte integer data types, reducing the bandwidth requirement to the host memory subsystem and to the rendering pipeline.

The SGI Onyx 3000 series also supports vertex arrays, which allow the application to issue primitive data to the Geometry Engine processors in larger transfers and with lower overhead compared to the standard fine-grain OpenGL interface. The SGI Onyx 3000 series extends the original OpenGL vertex array extension with support for packed vertex arrays, helping vertex arrays to be dispatched more efficiently while permitting tight packing of vertex coordinates and associated attributes.

7.3.2 Display List Storage Cache and Host Memory

Display lists are an excellent way of ensuring that data is being efficiently delivered to a rendering pipeline. InfiniteReality3 graphics support two features for substantially improved display list performance over immediate mode performance.

The first feature is a large display list storage cache built into the rendering pipeline. This cache is approximately 14MB and can deliver data to the geometry subsystem at more than 300MB/second with very low latency. The cache is automatically managed by a combination of OpenGL Performer and the OpenGL library. Display lists are selected for placement in the cache based on their size and their content. For example, very short lists will not be put in the cache since the overhead for executing them would outweigh the benefit of placing them in the cache. Lists containing a complex hierarchy will be excluded from the cache as the hierarchy traversal is too complex to deal with directly in the graphics pipeline.

The second display list feature allows display lists to be stored in host memory, but accessed directly by the graphics pipeline. As display lists are compiled they are stored in host memory in a representation suitable for direct interpretation by the Geometry Engine processor. When display lists are executed, the data does not need to be processed by the host CPU nor does it pass through any of the host CPU system caches. On SGI Onyx 3000 series systems, as with Silicon Graphics® Onyx2®, the bandwidth for these direct access display lists is approximately 320MB/second. This bandwidth is pipe-limited, not host-limited.

7.3.3 Primitive Processing

The SGI Onyx 3000 series has features for helping applications make better use of the Geometry Engine processors. One feature is the ability to perform per-vertex processing, such as viewing and lighting computations, on multiple primitives in one group. When primitives such as triangle strips are issued by the application, the vertices are accumulated and processed in batches. Geometry Engine processors and associated firmware are optimized for a particular batch length, usually around 6 to 12 vertices. The SGI Onyx 3000 series can process multiple primitives within the same batch, so it does not incur performance degradations.

7.3.4 Sprites

Visual simulation applications frequently need to perform computationally intense viewing transformations for drawing sprite objects, such as trees, in which the object rotates as the view changes. The SGI Onyx 3000 series provides a sprite OpenGL extension for performing these complex calculations in the Geometry Engine processors, thus freeing up the host processor for other computations.

7.4 Color

All OpenGL color and lighting models are supported within the architecture of the SGI Onyx 3000 series. Lighting operations are performed in floating point by the Geometry Engine processors. Color rasterization operations are performed by the raster subsystem with 48 bits (12 bits each for R, G, B, and A). Color index operations are supported at up to 12 bits per index. Colors may be stored in the frame buffer memory in visuals with as few as 16 bits or as many as 48 bits per color. More bits per color results in greater color fidelity and dynamic range. Special luminance formats allow the display and processing of monochrome images and textures with 16-bit components.

7.4.1 Shading

There is no penalty for smooth shading on objects rendered with InfiniteReality3 graphics. The Gouraud shading algorithm shades the surface of each polygon smoothly using the current lighting model. This yields clean, nonfaceted rendering of complex objects in real time and may be combined with transparency and texture for even more advanced rendering effects.

7.4.2 Color Blending

When multiple occluding surfaces have partial transparency, the raster subsystem can also perform color blending between each set of surfaces, depending upon the alpha values associated with each pixel. These alpha values commonly represent a degree of translucency from zero to one and enable simulation of windows, cockpit canopies, smoke, clouds, and other effects. To allow a wider choice of blending equations, a number of OpenGL blending extensions are supported.

7.4.3 Lighting

Hardware lighting capabilities enable the generation of more realistic-looking scenes. Lighting, which is computed per vertex, works with other rendering capabilities, including texture mapping, transparency, and Gouraud shading. All OpenGL lighting features and operations are supported.

7.4.4 Advanced Lighting Model

For creating realistic lighting scenarios, lighting effects must have many different components. Lights may be colored to allow for effects such as time-of-day lighting, where the color of the light as well as the intensity changes over time. Objects being lit may be affected both directly and indirectly by light sources. The OpenGL lighting model on SGI Onyx 3000 series systems provides for these characteristics in such a way that objects being lit are affected by a combination of the components shown in Table 2.

Table 2. Lighting Model Components

Lighting Model Component	Use
Specular	For highlights and simulating shiny surfaces
Diffuse	For broader, directional effects and simulating rough surfaces
Ambient	For environmental, nondirectional effects
Emissive	For self-luminous surfaces, vectors, or points

OpenGL commands are used to set up the current lighting model, affecting all subsequent surfaces sent down the pipeline until the model is changed. Effects of the components are combined to modify the color of each vertex sent to the graphics pipeline. Lights change the surface color based upon the four components of the current lighting model, the color of the light, and the defined surface properties.

7.4.5 Surface Properties

In addition to the four lighting model components, a shininess component may be set for the current lighting model to handle different surface properties. This component may be changed from object to object to allow for a variety of different surface types in the environment. This property, in addition to light characteristics and surface color, may be combined with texture mapping and transparency to provide the maximum flexibility in defining surface reflectance characteristics.

7.4.6 Infinite Light Sources

Infinite light sources are assumed to be infinitely far from the surface they are illuminating; thus, light rays from these sources are assumed to arrive in parallel at all the surfaces affected.

7.4.7 Local Light Sources

Local light sources may exist close to the surfaces they are illuminating; thus, the angle between the light source and the normals of the affected surfaces can change dramatically from one surface to the next. Properties such as cutoff angle and attenuation may be specified for each light.

Since angles from surface normals to the light source must be computed every frame for each surface, local lights are far more computationally expensive than infinite lights and should be used judiciously in real-time applications.

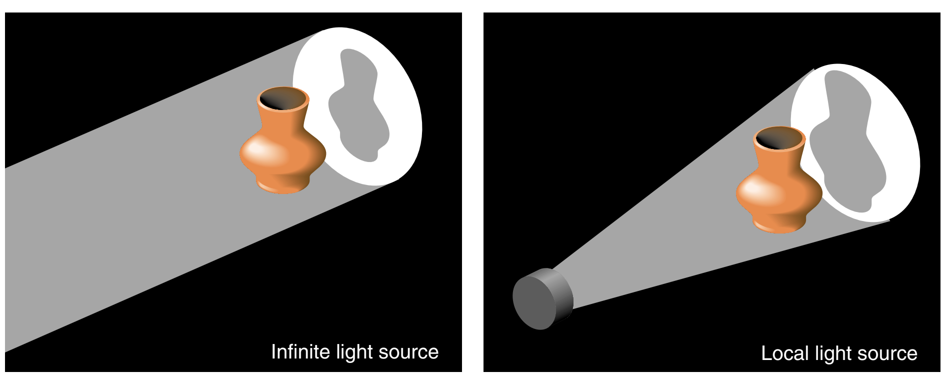


Figure 8. Infinite and Local Light Sources

7.5 Transparency

Each vertex to be rendered may have a transparency component, alpha, along with the standard color components: red, green, and blue. When alpha is used to control the pixel color, objects of varying transparency or translucency may be included in a rendered scene. To ensure that transparency is computed correctly and predictably, it is best to sort transparent objects from front to back with respect to the eyepoint, then draw these objects after all the nontransparent objects have been rendered.

7.6 Hidden Surface Removal

Current SGI machines use a Z-buffering technique to remove hidden surfaces. The Z (depth) buffer gives high-quality hidden surface removal during rendering without sorting the polygons. This process can thus be performed with few or no programming considerations in the software that renders the scene.

The SGI floating-point Z-buffer technique for SGI Onyx 3000 series systems advances the state of the art by offering more uniform resolution across the Z dimension. Unlike traditional floating point, which concentrates resolution closer to the eyepoint, SGI Onyx 3000 series systems distribute useful resolution throughout the viewing volume. SGI Onyx 3000 series systems offer three options for Z-buffer precision: 23-bit, 16-bit compressed, and 15-bit compressed. The Z-buffer exists as dedicated memory in the frame buffer associated with each pixel. The first time a pixel is drawn, its Z-buffer value is set to the Z of the primitive being drawn. Each time a subsequent attempt is made to set the color of that pixel because a new polygon is covering it, the system checks to see if that pixel's Z-buffer value is further or closer than the Z value of the pixel being drawn. If the new pixel is closer, its color value replaces the previous value in the color buffer and the Z-buffer is set to the new depth.

7.7 Coplanar Geometry Prioritization

There are several methods that may be used to support the rendering of multiple geometric layers of polygons that are coplanar. This situation arises when additional detail such as text, decals, or other markings needs to be added to a base polygon. With a conventional Z-buffering scheme, coplanar surfaces might Z-fight because the hardware cannot decide which face has visual priority; for example, in the case of runway markings the hardware cannot decide whether the markings are on top of the runway or vice versa.

SGI Onyx 3000 series systems support three methods of quickly assigning visible priority in these situations. First, with a combination of depth buffer offsets and depth buffer write-masking, a subface depth may be offset after projection without other visible effects and therefore assume visibility priority. This offset increases with the slope of the polygon to match and correct the characteristics of the depth buffer Z-fighting.

Second, fast stencil operations in conjunction with frame buffer stencil bits allow base polygons to be marked in the frame buffer stencil planes where visible. Subsequent rendering of subfaces tests for the base polygon stencil value instead of the Z-buffer value, which allows perfect subface rendering without any undesirable artifacts.

The final method of setting visible priority supported on SGI Onyx 3000 series systems allows subfaces to be rendered with Z-buffer information generated from the plane equation of the base polygon. The decal pixel fragments exactly match the base polygon depth values and therefore avoid any Z-buffer conflicts, giving the subfaces visible priority over the base polygon.

7.8 Anti-Aliasing

The high-performance anti-aliasing hardware of SGI Onyx 3000 series systems continues the trend started with the RealityEngine2™ graphics architecture. For the highest available image quality, memory and processors are put in the RM subsystem, resulting in multi-sampled anti-aliasing without the usual performance penalties. This anti-aliasing technique needs no data sorting and works with the Z-buffer for superior hidden surface removal.

Subpixel information is retained for all vertices in the pipeline as they are transformed and converted to raster primitives. Each primitive is computed with 8x8 subpixel accuracy. Thus, there are 64 possible subpixel locations within each pixel rendered. When deciding how to color a pixel, the raster system samples each primitive at one, four, or eight of the subpixel locations in each pixel it touches. It then calculates color and depth information for the sub-pixels covered. This information is sent to the frame buffer where Z-buffering is performed on each subpixel, and a final color for each pixel is formed from the accumulated data from the set of subpixels within it.

In multisample anti-aliasing mode, Z information is kept for every subsample, giving hidden-surface accuracy down to the subpixel level. When transparency is used, the alpha value is used to determine how much to blend the color of the closer pixel with the color of the farther pixel at that same location in the subpixel grid.

7.8.1 Sample Memory

The number of anti-aliasing samples available depends upon the total amount of frame buffer memory, the screen resolution, and the number of bits per image component desired. Table 3 shows some of the possible memory configurations for the SGI Onyx 3000 series with InfiniteReality3 graphics.

Table 3. Anti-Aliasing Sample Memory Options for InfiniteReality3 Graphics

One RM provides sufficient memory for:				
Medium pixels, 512 bits/pixel	One 960x680 display	8 samples	10 bits/component for RGB	16-bit Z-buffer, 8-bit stencil
Medium pixels, 512 bits/pixel	One 1280x1024 display	8 samples	10 bits/component for RGB	23-bit Z-buffer*, 1-bit stencil
Small pixels, 256 bits/pixel	Two 1280x1024 displays	4 samples	10 bits/component for RGBA	15-bit Z-buffer*, 1-bit stencil
Small pixels, 256 bits/pixel	One 1920x1080 display	4 samples	10 bits/component for RGB	15-bit Z-buffer*, 1-bit stencil
Two RMs provide sufficient memory for:				
Medium pixels, 512 bits/pixel	One 1280x1024 display	8 samples	10 bits/component for RGB	16-bit Z-buffer, 8-bit stencil
Medium pixels, 512 bits/pixel	One 1600x1200 display or one 1920x1080 display	8 samples	10 bits/component for RGB	23-bit Z-buffer*, 1-bit stencil
		4 samples	12 bits/component for RGBA	23-bit Z-buffer*, 8-bit stencil
Small pixels, 256 bits/pixel	Two 1920x1080 displays	4 samples	10 bits/component for RGB	15-bit Z-buffer*, 1-bit stencil
Four RMs provide sufficient memory for:				
Medium pixels, 512 bits/pixel	Eight 800x600 displays	8 samples	10 bits/component for RGB	23-bit Z-buffer*, 1-bit stencil
Medium pixels, 512 bits/pixel	Three 1280x1024 displays	8 samples	10 bits/component for RGB	23-bit Z-buffer*, 1-bit stencil
Medium pixels, 512 bits/pixel	Two 1920x1080 displays	8 samples	10 bits/component for RGB	23-bit Z-buffer*, 1-bit stencil
Medium pixels, 512 bits/pixel	One 1920x1200 display	8 samples	10 bits/component for RGB	16-bit Z-buffer, 8-bit stencil

*A 23-bit or 15-bit SGI Z-buffer implementation is roughly equivalent to a classic 32-bit or 24-bit Z-buffer, respectively.

7.9 Texture/Image Mapping

Designed to display complex, texture-mapped scenes at real-time frame rates (60 Hz), SGI Onyx 3000 series systems with InfiniteReality3 graphics remain compatible with earlier Onyx2 Reality™ and Onyx2 InfiniteReality2 systems while dramatically improving host performance and scalability.

7.10 Texture Download

SGI Onyx 3000 series systems download and draw texture simultaneously, allowing textures to be updated on-the-fly. Synchronization, built into the hardware and software, prevents textures from being replaced until the system finishes with them and prevents a texture being drawn until it is completely loaded. The maximum texture download rate is 330MB/second from host memory or 240MB/second from frame buffer. Unlike host download operations, the bottleneck for the texture readback from the frame buffer is the pixel path from the RM boards to the Geometry Engine board and is influenced by the type of visual used. See Table 4.

Table 4. Texture Rate for 1024x1024 Texture (RGB10 Visual for Framebuffer Read)

Source	16-bit	32-bit	48-bit
Host Memory	144 M texels/second	80 M texels/second	42 M texels/second
Frame Buffer Memory	85 M texels/second	85 M texels/second	47 M texels/second

7.10.1 Texture Formats

SGI Onyx 3000 series systems support three basic texture element (texel) sizes: 16-bit, 32-bit, and 48-bit, including 16-bit monochrome. Each size can represent several formats:

16-bit textures

- Dual 4-bit luminance with 4-bit alpha
- Quad 4-bit luminance, intensity, or alpha
- Dual 8-bit luminance, intensity, or alpha
- 8-bit luminance with 8-bit alpha
- 12-bit luminance, intensity, or alpha
- 12-bit luminance with 4-bit alpha
- 5-bit (each) RGB with 1-bit alpha
- 4-bit (each) RGBA
- 16-bit luminance, intensity, or alpha

32-bit textures

- 16-bit luminance with 16-bit alpha
- 12-bit luminance with 12-bit alpha
- 8-bit (each) RGBA
- 10-bit (each) RGB with 2-bit alpha

48-bit textures

- 12-bit (each) RGB
- 12-bit (each) RGBA

The variety of formats allows you to trade off image quality against storage size and rendering speed; 32-bit textures require twice the storage of 16-bit textures, while 48-bit textures require four times the storage of 16-bit textures. Generally, 16-bit textures render faster than 32-bit textures and 32-bit textures render faster than 48-bit textures for a given filtering mode.

The Texture Select feature is useful when vast amounts of low-resolution texture are required. With Texture Select, four 4-bit textures or two 8-bit textures are packed into one 16-bit texture. The user may select which of the four (or two) textures are used at any given time.

7.10.2 Texture Filtering

One of the most powerful features of the architecture for the SGI Onyx 3000 series with InfiniteReality3 graphics is the range of advanced two-dimensional and three-dimensional texture filtering techniques that are supported. This variety allows you to choose the most appropriate filtering technique for your application.

The filtering techniques comprise two different classes: MIPmapped and non-MIPmapped. The term MIPmapping is derived from the Latin phrase, multum in parvo, which means many things in a small space. The MIPmapping process takes a texture (image) to be used in a scene and generates a series of prefiltered, lower-resolution versions of that texture. The appropriate level of the MIPmapped texture is then applied to a polygon as required by the scene. This allows the texture to be minified without aliasing artifacts. Different MIPmapped filtering techniques are then available to calculate the transitions between different MIPmap levels and between different texels (texture elements) within a level. These filtering techniques include:

- Point sampled
- Linear interpolated
- Bilinear interpolated
- Trilinear interpolated
- Quadlinear interpolated (for 3D textures)

Non-MIPmapped textures require filtering only among the different texels within the texture map. These filtering techniques include:

- Point sampled (for 2D or 3D textures)
- Bilinear interpolated (for 2D textures)
- Trilinear interpolated (for 3D textures)
- Bicubic interpolated (for 2D textures)

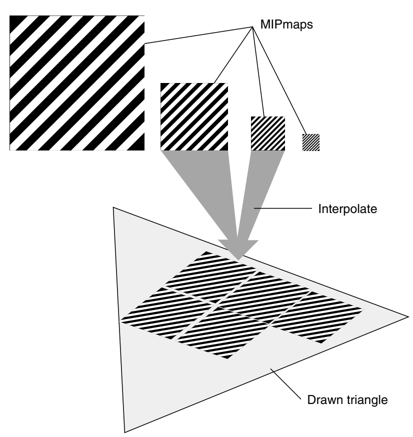


Figure 9. MIPmapping

Except for quadlinear interpolated MIPmapping and bicubic interpolated filtering, which operate at half the rate of the others, the SGI Onyx 3000 Series with InfiniteReality3 graphics is designed to provide all of the filtering techniques listed above at full texture processing performance levels. It is important to recognize that trilinear interpolated non-MIPmapped 3D texture filtering is available at full texture processing performance levels.

7.10.3 Global Texturing a.k.a. Clipmapping

Global texturing accelerates processing of terrain data mapped with satellite and aerial photographs. Global texturing allows you to define single MIPmap images which are much larger than that which can be stored in texture memory. Only the parts of the MIPmap that are actually used in rendering an image are stored in texture memory; the rest is stored in main memory or on disk. As the point of view moves, the portion of the MIPmap that is stored in memory is updated to reflect the new point of view.

The implementation is effective for real-time simulation because the paging of texture is performed in anticipation of the requirements of the visual scene. These requirements are directed by the eye position before the texture is required. This avoids the problems of paging on demand that would stall the graphics system while waiting for essential texture information from system memory or disk. At the same time the terrain is easily modeled with just a single texture image used to represent the entire database without the need for tiling.

The paging is performed in real time for each level of MIP in the image pyramid that is deemed too large to fit in texture memory. So one can imagine a requisite subset of each image in the MIPmap pyramid cached in texture memory; let's say we store a 1024x1024 image for each MIP texture image (see Figure 10). If our base image held 1 meter resolution data then the texture hardware holds 1 square kilometer of information at highest resolution. When clipmapping, the next level of MIP would also hold a 1024x1024 resolution image cache, but because this MIP level is lower resolution that same amount of data covers a 2-square-kilometer database area with 2 m texel data. This continues for each successive level of MIP until the entire texture image is held in texture memory at some ground resolution; this would be where a 1024x1024 image held the whole geographic database. Each 1024x1024 image held in memory covers a concentric region of interest around some important point in the database; normally this point is the center of the area the graphics system is currently rendering at highest detail for this frame. As the eye moves over the database the trailing border of each image cache is replaced by the information required by the leading edge, and the texture addressing is torroidally displaced to account for the newly downloaded image data (see Figure 11).

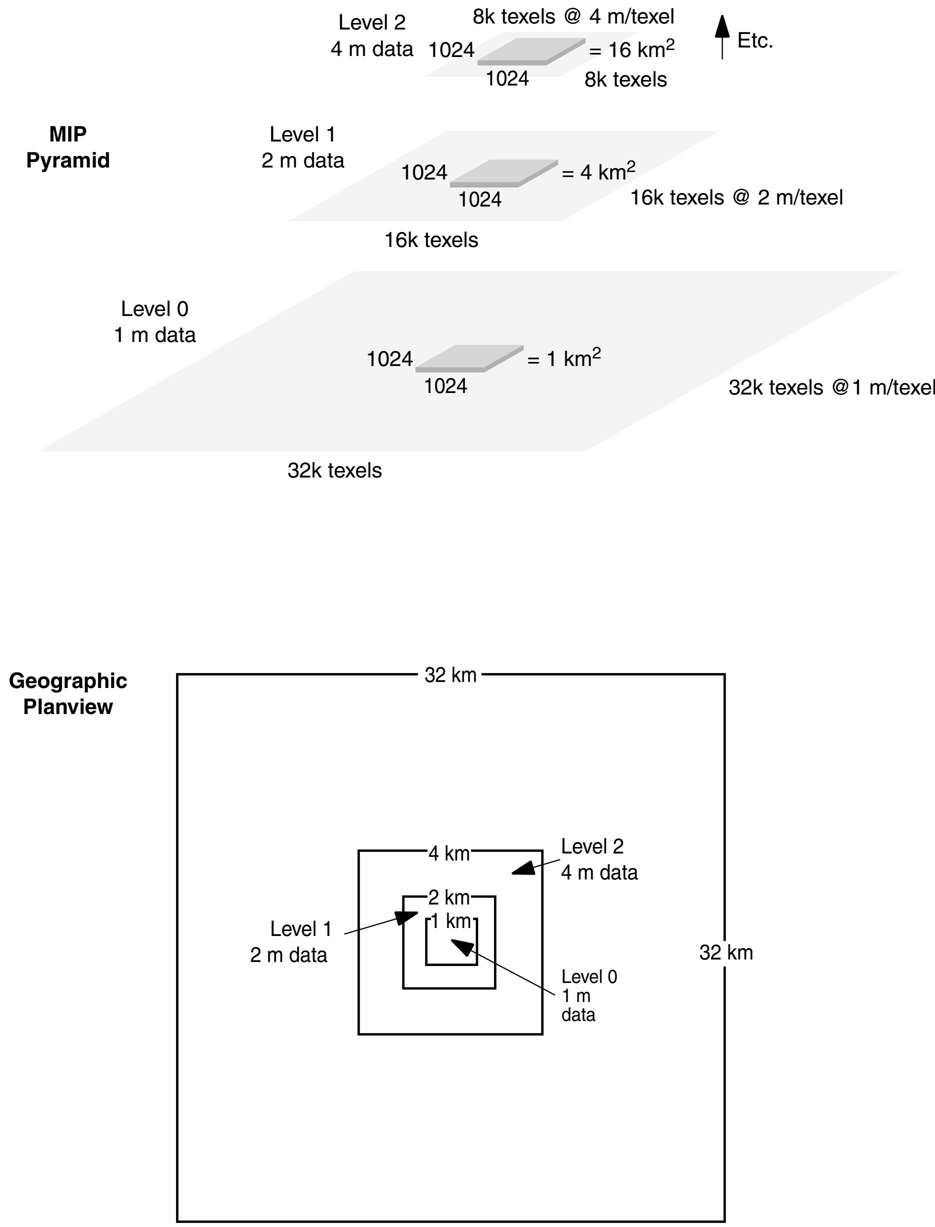


Figure 10. Clipmapping

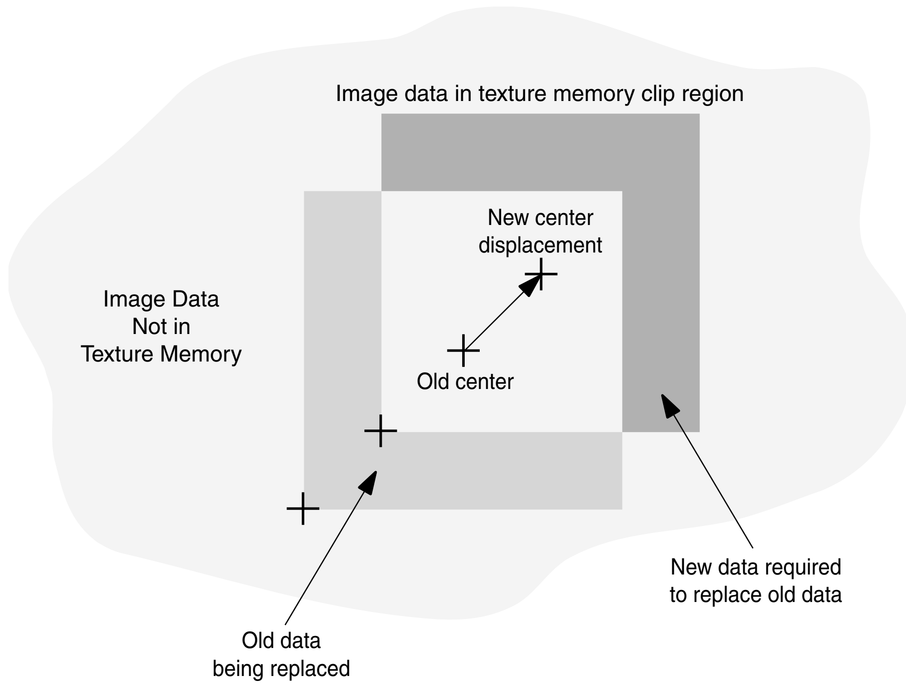


Figure 11. Clipmapping

7.10.4 Add, Replace, Blend, Decal, and Modulation

Each texture can be treated in several ways when applied to a polygon. When operating in add texture mode, the texture color is added to the result of the lighting calculation or polygon color. A color multiplier can be used for added flexibility. With replace or decal mode, the color of the texture replaces the inherent color of the polygon; alpha treatment is the difference between these two modes. In replace mode, the texture color replaces the inherent color of the polygon. When using an RGBA texture, the texture alpha is the resultant alpha; with an RGB texture, the polygon alpha is the resultant alpha. Blend mode uses the texture color as a blending value for the mixing of the polygon color and any specified blend color. You can use a texture image as a matte between the polygon color or lighting result and any other color. During modulation, the texture color is multiplied by the lighting result or polygon color to yield the classic textured lit appearance seen in most graphics images.

7.10.5 Texture/Image Memory

Each RM in the InfiniteReality3 graphics system has 256MB of texture memory. RMs in the InfiniteReality2 graphics system have 64MB of texture memory, while those in the Reality graphics system have 16MB. For non-MIPmapped textures this means that the texture memory can hold 96 million, 32 million, or 8 million 16-bit texels, respectively. MIPmapped textures require extra storage for the prefiltered lower levels of detail, amounting to one-third the size of the finest level of detail for 2D and one-seventh the size of the finest level of detail for 3D.

Texture memory is divided into two banks for performance reasons. Because of this, the maximum size texture does not always fill up texture memory. With the exception of bicubic, an image, a level of a MIPmap, or slices of a volume cannot cross a texture memory bank. The largest 16-bit texture maps that fit into SGI Onyx 3000 series systems with InfiniteReality3 graphics texture memory are shown in Table 5. Other aspect ratios with the same number of pixels will also fit.

Table 5. Largest Available 16-Bit Texture Maps

Texture Memory	2D Non-MIPmap	2D MIPmap	2D Bicubic	3D Non-MIPmap
256MB	4096x4096	2048x2048	4096x4096	512x512x256
64MB	4096x4096	2048x2048	4096x4096	512x256x256

Table 6 shows the number of 16-bit textures that can be stored in 256MB of texture memory. Table 7 lists the same information for several 3D textures.

Table 6. 2D Texture Memory Sizes and Quantities

Texture Size	Number of 16-Bit MIPmapped Textures in 256MB of Texture Memory	Number of 16-Bit Non-MIPmapped Textures in 256MB of Texture Memory
2048x2048	24	32
1024x1024	96	128
512x512	384	512
256x256	1,536	2,048
128x128	6,144	8,192
64x64	24,576	32,768
32x32	98,304	131,072
16x16	393,216	524,288

Table 7. 3D Texture Sizes for 16-Bit Texture

Texture Memory	Resolution	Number of 16-Bit Textures
16MB	128x128x128	4 (non-MIPmapped)
16MB	128x128x128	2 (MIPmapped)
16MB	256x256x128	1 (non-MIPmapped)
16MB	256x128x128	1 (MIPmapped)
64MB	256x256x256	2 (non-MIPmapped)
64MB	256x128x128	7 (MIPmapped)
64MB	256x256x512	1 (non-MIPmapped)
64MB	256x256x256	1 (MIPmapped)
256MB	512x512x256	2 (non-MIPmapped)
256MB	512x512x256	1 (MIPmapped)

7.10.6 Texture Memory Packing

Although supported texel sizes pad to 16-bit sizes, InfiniteReality3 graphics support the packing of multiple smaller format images into the same memory space; for example, an image with 16-bit texels can pack two 8-bit images into a single 16-bit-per-texel image. This feature allows the efficient use of texture memory for small format images, particularly single component images.

7.10.7 Texture Transparency and Contouring

Textures may have full or partial transparency set at any texel location. If the entire outer edge of a uniquely shaped texture, such as a tree, is set to transparent, the texture may be placed upon a rectangular polygon and still appear to have the outline of a tree.

7.10.8 Perspective Correction

The raster subsystem performs per-pixel computations to correct textures and fog for perspective distortions to ensure that no artifacts are introduced into the rendering process. These computations do not affect system performance.

7.10.9 Detail Texture

Detail texture increases the effectiveness of motion and position cues by providing additional detail on top of the existing texture as the textured surface draws nearer to the eyepoint. Detail texture allows high-resolution textures to be represented with a small high-frequency characteristic texture and a lower resolution base texture. The high-frequency detail is smoothly combined with the base texture to give much of the visual qualities of the original texture, while using a fraction of the texture memory.

7.10.10 Sharp Texture

Textures enable creation of highly realistic scenes, largely due to the ability to import photographic images, which can be applied to geometry. The resolution of these images is finite, however. When these images are enlarged, magnification tends to blur the original image.

The solution is the patented sharp texture feature in the InfiniteReality3 graphics of the SGI Onyx 3000 series, which lets you specify that edges remain sharp during magnification. For example, this enables the development of a simulation whose photo-derived sign textures would continue to be readable as they get close to the viewer's eyepoint.

7.10.11 Projective Texture

With projective texture, textures can be projected into a scene, bending over every surface. This allows for sophisticated per-pixel spot-lighting effects, such as landing lights, and projection of camera data onto synthetic terrain.

7.10.12 Shadow Maps

SGI Onyx 3000 series systems with InfiniteReality3 graphics provide hardware support for shadowing. Applying shadows is a two-step process: the scene of interest is first rendered from the light source point of view; the depth information from the scene is then copied into texture memory. The result is a shadow map. The scene is then rendered from the eyepoint. Hardware determines which parts of the scene are in shadow and which are not, based on the shadow map, and lights them accordingly. The approach can be extended to multiple light sources by rendering in multiple passes.

7.10.13 Billboards

Billboards are textured faces that automatically track the eyepoint so that they are always facing the viewer. The billboard feature is commonly used for trees and other complex but approximately symmetric objects.

7.10.14 Detail Management

You can restrict which MIPmap levels of detail are used. This restriction allows use of a partially loaded MIPmap. For example, if a MIPmap is loaded starting at the coarsest level, detail management allows the texture to be used before its finest levels of detail have been loaded.

7.10.15 Video to Texture

With either the Digital Video Option or the High-Definition Digital Video Option, broadcast-quality live video may be brought into host memory in real time with very low latency and in formats compatible with OpenGL. This image data stream may then be sent to texture or raster memory for processing by the graphics pipeline. This provides the capability to perform true three-dimensional (3D) texture mapping with a live video source. In addition, image processing or other manipulations may be performed on the video data for image analysis or video effects.

7.10.16 3D Textures and Volume Rendering

SGI Onyx 3000 series systems include the advanced 3D texture feature, which enables real-time 3D volume rendering. For example, 3D texture could be used to view a series of two-dimensional medical scans of a patient's head and to reconstruct these scans into a three-dimensional volume. The volume can be manipulated (rotated, translated, filtered, and more) in real time by continuously resampling (interpolating) the data in 3D. Volume rendering can be enhanced using the following features:

- Slicing planes
- Perspective projection
- Embedded surfaces

For more information, see Fraser, Robert, “Interactive Volume Rendering Using Advanced Graphics Architectures,” Silicon Graphics Computer Systems.

7.10.17 Texture Color Lookup Tables

SGI Onyx 3000 series systems support texture color lookup tables without any performance penalty. This table can change in real time to achieve selection in volume rendering or transfer functions for sensor simulation. The hardware supports both monochrome and color functions and has 12-bit precision.

7.11 Imaging Operations

The Geometry Engine processors of the InfiniteReality3 graphics system provide hardware accelerated imaging operations, including convolution, minmax, histogram, color matrix, and lookup tables. Because all pixel processing paths, including drawing, copying, and reading of pixels, texels, and video data, flow through the Geometry Engine processors, these image operators can be applied between any source and destination. For example, an application may copy pixels from a video source to texture memory and perform colorspace conversion using the color matrix in one pixel transfer operation.

7.11.1 Convolution

The convolution operator filters linear images in real time. The user-definable filter performs operations such as sharpening and blurring. SGI Onyx 3000 series systems support separable and general convolution filters of size 3x3, 5x5, and 7x7. Separable filters are much faster than general filters. Likewise, smaller filters are faster than larger filters. Applications can trade off quality and throughput by choosing an appropriate filter size and type.

7.11.2 Histogram and Minmax

Histograms and minmax provide statistics on color component values within an image. A histogram counts occurrences of specific color component values and reports them in user-defined bins with up to 4,096 bins per component. Minmax tracks minimum and maximum color component values. These statistics can be analyzed to create a more balanced, better-quality image.

7.11.3 Color Matrix and Linear Color Space Conversion

The color matrix operation transforms the colors of each pixel by a user-defined 4x4 matrix. In addition to performing linear colorspace conversion, the color matrix can also be used to perform color component swapping and component replication.

7.11.4 Window/Level Support

Remote sensing, medical imaging, and film editing applications can process data with a greater dynamic range than a video monitor can display. The window/level operation allows a subset of the datas full dynamic range to be selected (windowed) and its contrast and brightness adjusted (leveled) so that important image features will be seen on a monitor.

The frame buffer and imaging data path of SGI Onyx 3000 series systems support display and texturing using 16-bit luminance data, which is appropriate for demanding, high-precision imaging applications. The 32K-entry color lookup table on the Display Generator board enables real-time window/level operations on multiple windows over 12-bit subranges without requiring images to be redrawn.

7.11.5 Lookup Tables

Color tables can be inserted in various stages of the pixel path to adjust image contrast and brightness after an image operation. Each color table can have up to 4,096 entries per color component and can be loaded directly from host or frame buffer memory.

7.12 Atmospheric Effects

A programmable fog/haze function in SGI Onyx 3000 series systems allows setting an arbitrary fog color that can be blended with the color of any element to be rendered. The fog may be changed for all or part of the scene on a per-channel or per-object basis. The amount of fog or haze color used depends on the depth value of each pixel of each selected object.

The user can specify the falloff function and the density of the fog. SGI Onyx 3000 series systems also allow specifying a fog function using sample points.

7.13 Offscreen Rendering

SGI Onyx 3000 series systems have abundant frame buffer memory with 80MB on each InfiniteReality3 RM. Frame buffer memory stores pixels for very-high-resolution displays or for multiple video output channels. It also stores pixels for offscreen rendering.

A frame buffer pixel contains from 256 to 1,024 bits, depending upon the number of RMs in the system and video output channel characteristics. Some bits are dedicated to front and back color buffers (when applications use double buffering), the Z (depth) buffer, and other buffers supported by OpenGL (see Figure 12).

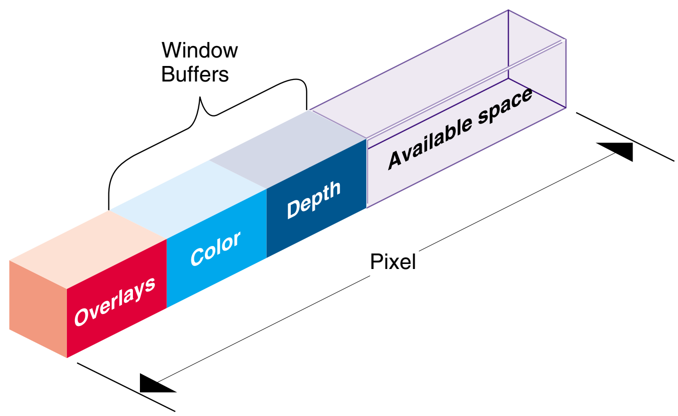


Figure 12. Frame Buffer Pixel Memory

Remaining memory is available for other uses. Applications may invoke the SGIX_fbconfig OpenGL extension to choose a frame buffer configuration for unassigned memory (see Figure 13).

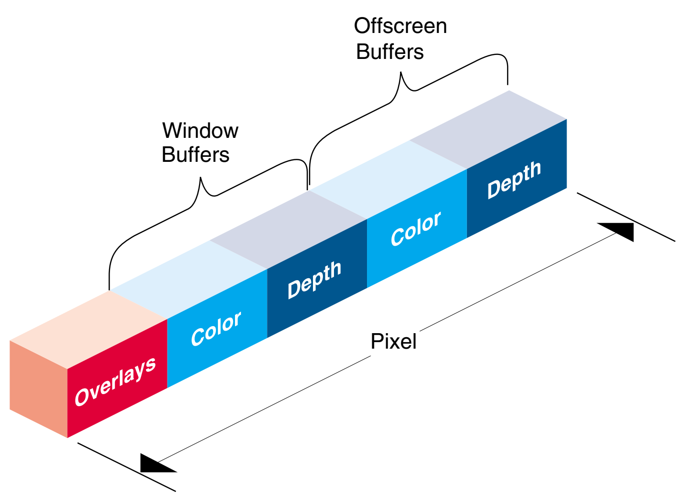


Figure 13. Frame Buffer Pixel Memory Including Offscreen Buffers

After selecting a frame buffer configuration, applications may invoke the SGIX_PBuffer OpenGL extension to create an offscreen rendering area called a PBuffer (Pixel Buffer). PBuffers occupy part of the frame buffer not otherwise occupied by ordinary windows (see Figure 14).

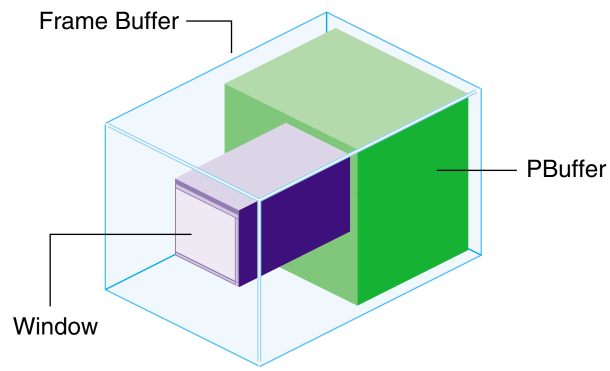


Figure 14. Disjoint Buffers and PBuffers

PBuffers are much like windows; OpenGL applications may render into PBuffers just as they render into windows. However, PBuffers never obscure color buffers of windows and are never displayed on any video output channel.

PBuffers may serve as a cache for frequently accessed images, improving performance dramatically by removing the need to transfer pixels from host memory to the graphics pipeline. They may be used for storing results of special-purpose renderings that are not intended for display, such as shadow maps. PBuffers are also convenient repositories for the intermediate results of multipass rendering algorithms, which produce effects such as reflections, Phong shading, and bump mapping. Finally, PBuffers permit SGI Onyx 3000 series systems to accelerate background batch processing of long-running imaging operations without disabling interactive use of the main display.

7.14 Multichannel Display Generator

The Display Generator takes digital, ordered pixel output from the frame buffer and specifies from one to eight separate rectangular areas to be sent from the rectangular frame buffer area managed by the X-server to independent component RGB video outputs. Each video channel may have its own video timing (subject to constraints detailed in Section 7.14.11), allowing for maximum flexibility.

7.14.1 Standard Formats and Format Combinations

To provide as much built-in flexibility as possible, SGI Onyx 3000 series systems with InfiniteReality3 graphics are delivered with a broad range of software-selectable video formats and format combinations, from NTSC to 1920x1080 progressive-scan HDTV. Table 8 and Table 9 show SGI Onyx 3000 series video formats and format combinations, respectively, shipped standard with the system.

Table 8. Standard Formats Shipped with All SGI Onyx 3000 Series Systems

Video Format	Video Format File	Notes
640x480 120 Hz stereo	640x480_120s.vfo	
640x480 180 Hz field sequential	640x480_180q.vfo	Color field sequential 180 Hz format
640x480 60 Hz	640x480_60.vfo	VGA
646x486 30 Hz interlaced	646x486_30if.vfo	Frame locking format
646x486 30 Hz interlaced	646x486_30i.vfo	NTSC
768x576 25 Hz interlaced	768x576_25i.vfo	PAL
768x576 25 Hz interlaced	768x576_25if.vfo	Frame locking format
800x600 60 Hz	800x600_60.vfo	SVGA
960x620 60 Hz	960x620_60.vfo	
960x680 60 Hz	960x680_60.vfo	
1024x768 120 Hz stereo	1024x768_120s.vfo	
1024x768 60 Hz	1024x768_60.vfo	
1024x768 96 Hz stereo	1024x768_96s.vfo	
1080x809 30 Hz interlaced	1080x809_30i.vfo	RS343-A “875-line” format
1120x840 96 Hz stereo	1120x840_96s.vfo	
1200x900 72 Hz	1200x900_72.vfo	High-resolution 4:3 format
1280x1024 114 Hz stereo	1280x1024_114s.vfo	
1280x1024 120 Hz stereo	1280x1024_120s.vfo	Ultrahigh-resolution stereo
1280x1024 25 Hz PAL	1280x1024_25r2.vfo	Special format
1280x1024 25 Hz PAL	1280x1024_25r3.vfo	Special format
1280x1024 30 Hz NTSC	1280x1024_30r2.vfo	Special format
1280x1024 50 Hz	1280x1024_50.vfo	
1280x1024 60 Hz	1280x1024_60.vfo	High-resolution
1280x1024 72 Hz	1280x1024_72.vfo	
1280x959 30 Hz interlaced	1280x959_30i.vfo	
1500x1200 60 Hz	1500x1200_60.vfo	
1600x1024 60 Hz	1600x1024_60.vfo	Silicon Graphics® 1600SW flat panel display Wide XGA resolution 16:10 format
1600x1200 60 Hz	1600x1200_60.vfo	Extra-high-resolution

Continued on page 42

Continued from page 41

Video Format	Video Format File	Notes
1760x1100 60 Hz	1760x1100_60.vfo	
1920x1035 30 Hz interlaced	1920x1035_30i.vfo	Japan HDTV
1920x1080 72 Hz	1920x1080_72.vfo	U.S. HDTV
1920x1200 66 Hz	1920x1200_66.vfo	Ultrahigh-resolution
CCIR601_525	CCIR601_525.vfo	
CCIR601_625	CCIR601_625.vfo	

Table 9. Standard Format Combinations Shipped with All SGI Onyx 3000 Series Systems

Standard Combinations	Combiner File
1 1280x1024 60 Hz	1280x1024_60.cmb
1 1280x1024 72 Hz	1280x1024_72.cmb
1 960x680 60 Hz	960x680_60.cmb
1 1024x768 60 Hz	1024x768_60.cmb
1 1200x900 72 Hz	1200x900_72.cmb
1 1600x1024 60 Hz	droid.cmb
1 1600x1200 60 Hz	1600x1200_60.cmb
1 1920x1035 30 Hz interlaced	1920x1035_30i.cmb
1 1920x1080 72 Hz	1920x1080_72.cmb
1 1920x1200 66 Hz	1920x1200_66.cmb
1 1280x1024 60 Hz and 1 NTSC	1280x1024 + NTSC_60.cmb
1 1280x1024 72 Hz and 1 1920x1080 72 Hz	1280x1024 + 1920x1080_72.cmb
1 1280x1024 60 Hz and 4 640x480 60 Hz	1280x1024 + 4@640x480_60.cmb
1 1280x907 60 Hz and 5 640x453 60 Hz	1280x907 + 5@640x453_60.cmb
2 1280x1024 60 Hz	2@1280x1024_60.cmb
2 1280x1024 72 Hz	2@1280x1024_72.cmb
3 1024x768 60 Hz and 1 1280x959 30 Hz interlaced	3@1024x768_60 + 1280x959_30i.cmb
3 1280x1024 60 Hz	3@1280x1024_60.cmb
1 640x480 120 Hz stereo	640x480_120s.cmb
7 800x600 60 Hz and 1 NTSC	7@800x600_60 + NTSC.cmb
8 640x480 60 Hz	8@640x480_60.cmb
8 800x600 60 Hz	8@800x600_60.cmb

7.14.2 2-Channel and 8-Channel Configurations

Both versions of the SGI Onyx 3000 series video display subsystem are multichannel-capable. The standard version has two independent video channels for driving two separate RGB displays or virtual reality stereoscopic displays or for a single RGB video channel and an independent video encoder. The high-performance version provides eight independent channels, each with its own 12-bit gamma table. On either version, the second channel can be a normal RGB channel or the video encoder channel.

Because multichannel capability is included in the standard system, users do not need to sacrifice a slot in the backplane to get this feature, even if the system is configured for eight outputs. Use of the InfiniteReality3 multichannel feature does not disturb or disable operation of the main video output channels. All additional channels are as easy to program as the main channel; the same video format compiler works for all channels, including the Serial Digital Video Channel enabled by the Graphics-to-Video Out (GVO) option and High-Definition Graphics-to-Video Out (HD GVO) option.

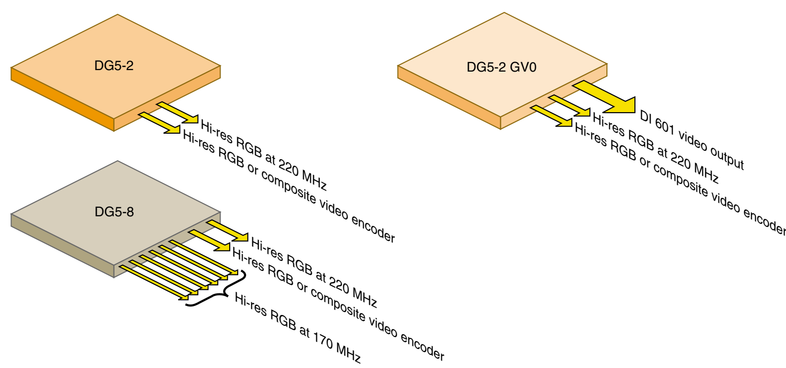


Figure 15. 2- and 8-Channel Options

7.14.3 SGI Onyx 3000 Series Multichannel Features: Considerations

Several system resources must be considered in multichannel applications: frame buffer memory, frame buffer read/write bandwidth, DAC bandwidth, and frame buffer-to-video subsystem transmission bandwidth.

7.14.3.1 Swap Rates Must Be Equal

Video formats that run together in a video combination must provide time for housekeeping operations such as updating cursor position and glyph, setting parameters for dynamic pixel resampling, and more. This may be done at field boundaries, frame boundaries, or in certain cases, across multiple frames. This interval is defined by the maximum swap rate. This rate must be the same for all video formats running together, so that the SGI Onyx 3000 series video display subsystem can simultaneously perform these housekeeping services for all running video formats. For example, a legal combination of formats would be the following:

- 30 Hz interlaced NTSC (swap rate is the 60 Hz field rate for this format)
- 60 Hz noninterlaced 1280x1024
- 120 Hz stereo (swap rate is the 60 Hz frame rate)
- 180 Hz color field sequential (again, the swap rate is the 60 Hz frame rate)

Examples of illegal combinations would be NTSC (60 Hz swap rate) and PAL (50 Hz swap rate) or 60 Hz and 72 Hz versions of 1280x1024. In these cases, the swap rates are not the same. Equal is a relative term. For example, NTSC, which really runs at 59.94 Hz field rates, is close enough to be run with 60 Hz 1280x1024. In general, if the swap rates are close enough to allow the video formats to be reliably synchronized to one another, they are considered equal. This tolerance is usually less than 0.2 percent.

Figure 16. Equal and Unequal Swap Rates

7.14.3.2 Frame Buffer to Video Display Subsystem Transmission Bandwidth

Digital video data is sent serially from the InfiniteReality3 frame buffer to the video subsystem. Each video channel uses a portion of the systems aggregate video transmission bandwidth. The required bandwidth per channel depends on the resolution of the video output format running on that channel and on the type of video it receives from the frame buffer. The requirements of all video channels must be less than the aggregate video bandwidth of the system.

Depending on the number and precision of color components requested from the frame buffer by each video channel, the available video transmission bandwidth ranges from approximately 290 M pixels/second (for 10-bit RGB or 12-bit color field-sequential video) to 210 M pixels/second (for 10-bit RGBA or 12-bit RGB video). The digital data stream from the frame buffer to a particular video channel must be of a single type and precision, for example, 10-bit RGB. However, different video channels (and associated digital video streams from the frame buffer) may be mixed: one channel may request 10-bit RGB, while another channel may request 12-bit RGB, and yet another channel may request 10-bit RGBA. Depending on the mix of currently running video formats, the total required video transmission bandwidth will fall between approximately 210 M and 290 M pixels/second.

Distinguishing the transmission format for video from the depth of pixels in the frame buffer is important. Pixel depth must be uniform across the entire managed area of the frame buffer. Pixel depths are defined as small (256 bits per pixel), medium (512 bits per pixel), large (1,024 bits per pixel), or extra-large (2,048 bits per pixel). As stated above, the format of the colors transmitted as video from a frame buffer to the video display subsystem may vary on a per-channel basis to conserve aggregate video transmission bandwidth. Frame buffer memory stores nonvideo information about each pixel (such as multisample and Z information); the color fields are a subset of the frame buffer pixel storage. Distinguishing between frame buffer representation of pixels and the format chosen for video transmission to the video display subsystem is important if you want to understand and optimize the performance of both the frame buffer memory and video display subsystem.

The total frame buffer to video subsystem bandwidth is independent of the number of RMs in the system.

7.14.3.3 DAC Output Bandwidth

When configuring a video format combination, you must decide which format to assign to which video channel. The first two video channels have DACs with a 220 M pixel/second bandwidth limit. The remaining six channels have DACs with a 190 M pixel/second bandwidth limit on the eight channel version. Because each channel can do dynamic resolution (see Section 7.14.7), actual video resolution as output by the DAC (and seen by the display monitor connected to the channel) may be higher than the resolution of the rectangular region of the frame buffer assigned to that channel. The highest-bandwidth video formats of a video format combination should always be assigned to the first two video channels, 0 and 1.

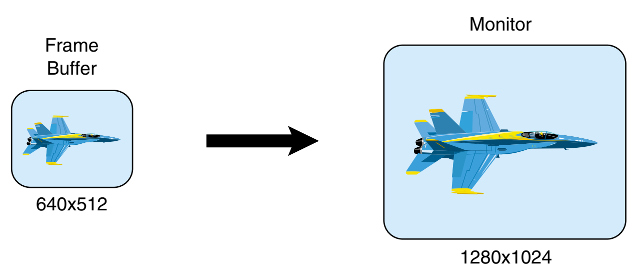


Figure 17. Dynamic Resolution

7.14.3.4 Frame Buffer Memory

Another requirement for video format combinations is that they must all be able to tile rectangular subregions of a rectangular frame buffer. Depending upon the depth of pixels selected and the number of RMs installed, some combinations of formats may require more than one RM. When selecting frame buffer pixel depth you must consider the quality of multisampling, the precision of the Z-buffer and RGB colors stored in the frame buffer, and total system fill rate capability.

Adding RMs does not increase total frame buffer to video subsystem bandwidth, but it increases the total drawing fill rate available and amortizes the video refresh overhead over more RMs, leaving more fill capacity available for drawing.

7.14.3.5 Frame Buffer Read/Write Bandwidth

A final consideration for video format combinations is the overhead that refreshing video channels presents to the frame buffer. The greater the number and resolution of video channels, the more the pixel fill rate of the system is reduced. A particular combination of formats may fit into a 1-RM or 2-RM configuration, but it may unacceptably reduce the fill rate.

7.14.4 Video Format Combiner

To make the necessary trade-offs between these requirements, the system includes the Video Format Combiner software tool. This tool has a command-line interface and a graphical user interface (GUI). With the GUI form, it is easy to see which regions of the frame buffer are mapped to which channels. Pixel bandwidth and fill rate overhead due to video are dynamically displayed. When system limits are exceeded, the tool attempts to provide hints about what remedial action to take.

If system resources of a target hardware configuration are sufficient to support a group of video formats, the combiner tool creates a combination file with filename references to the video formats and information about configuring the video hardware to run that channel combination.

Only video format combinations approved by the video format combiner are allowed to be loaded into the video subsystem (even if only a single channel is enabled). This is done with the video format combiner itself or with the familiar setmon program. In addition, there is a programming interface provided by an X-Windows™ server extension, XSGIvc, for application developers who want to integrate video control into their applications.

Figure 18 shows an example of using the video format combiner tool interactively to place six video channels on a 1-RM InfiniteReality3 graphics system in a managed area of 1920x1360 pixels. Small (256-bit) pixels are used. Channel 0 is a 1280x1024 video format mapped to a 1280x907 region of the frame buffer using static resizing. Channels 1 through 5 are in a 640x480 video format. With static resizing, each is mapped to a 640x453 region of the frame buffer. The digital media options are discussed in Section 7.18.

This example illustrates the use of video resizing to efficiently use the frame buffer. If static resizing were not used, only four 640x480 channels would fit in addition to the 1280x1024 channel. Note the indicators at the bottom of the screen that show this video format combination is using 58.90 percent of the available pixel transmission bandwidth and that the fill capability of this combination is reduced by 18.45 percent. For comparison, a single 1280x1024 screen would reduce the fill rate of a 1-RM system by about 10 percent. In other words, when this combination is run, 81.55 percent (100 percent minus 18.45 percent) of the fill rate is available for drawing into the frame buffer.

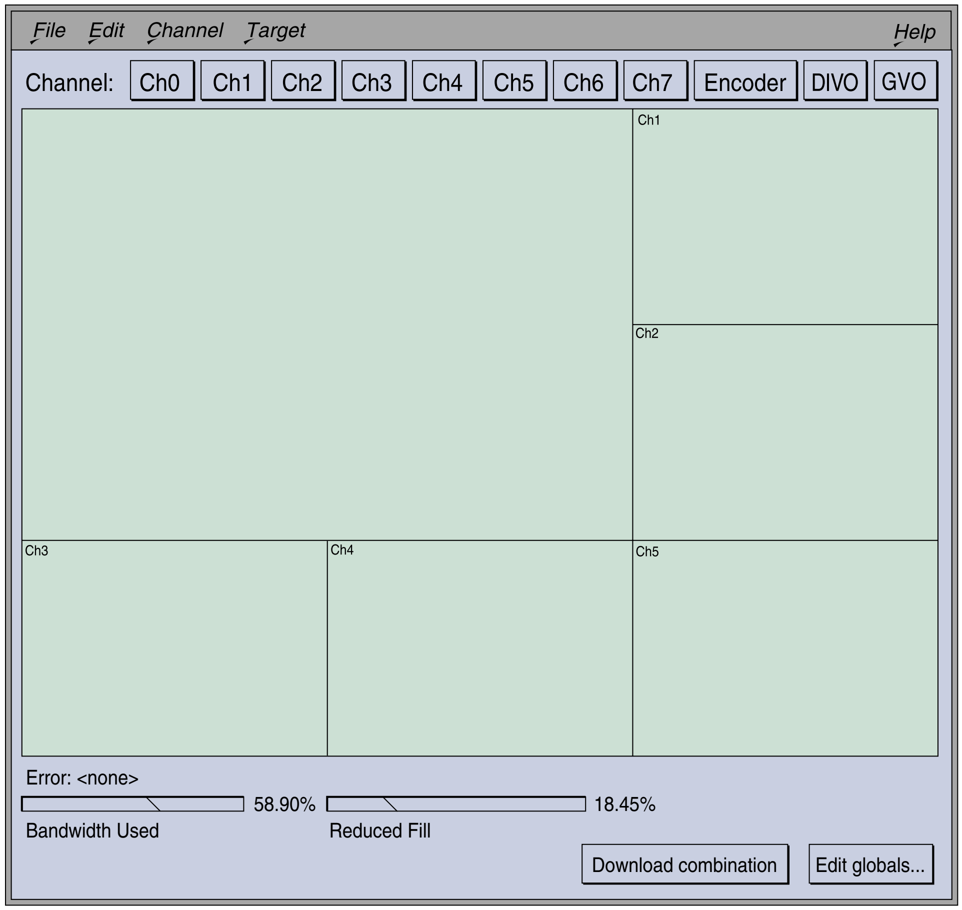


Figure 18. Using the Video Format Combiner to Place Six Video Channels

7.14.5 Examples of Video Format Combinations

As an example, a 1-RM system supports two 1280x1024, 60 Hz noninterlaced channels using small (256-bit) pixels. Adding another RM allows the use of two of these channels with medium (512-bit) pixels or three channels using small pixels. A 2-RM configuration supports eight channels at 800x600 pixels at 60 Hz. The video combiner tool allows these kinds of what-if scenarios to be explored, enabling users to easily configure SGI Onyx 3000 series systems for their particular applications. Table 10 shows several example configurations. Many others are possible.

Table 10. Examples of Video Format Combinations

Video Format Combinations	Notes
2 @ 1280x1024, 60 Hz	Uses all memory on 1 RM (small frame buffer pixels)
3 @ 1280x1024, 72 Hz	Uses nearly the entire available pixel bandwidth; requires 2 RMs for sufficient memory (small pixels)
8 @ 800x600, 60 Hz	Uses all channels; needs 2 RMs for sufficient memory (small pixels)
7 @ 800x600, 60 Hz + NTSC Encoder	Uses all channels; Channel 1 is used by Composite Video Encoder when enabled
3 @ 1024x768, 60 Hz + RS343-A 1260x945, 30 Hz interlaced	Not all formats need the same resolution; field rates of all video formats must match; requires 2 RMs for sufficient memory (small pixels)
1 @ 1920x1080 + 1 @ 1280x1024 both at 72 Hz	1920x1080 channel must be assigned to high-speed DAC (channel 0 or channel 1); requires 2 RMs for sufficient memory (small pixels)

7.14.6 Video Format Compiler

A video format is the set of electrical and timing characteristics that drive a monitor (or any video output device). In the context of this report, a video format usually refers to the video format source language used to describe the video format itself.

The video format compiler takes the video format source language file and produces a video format object file. You can use the video format object file to load video generation hardware (some hardware architectures require an intermediate step that combines multiple video format object files).

7.14.7 Dynamic Resolution

SGI Onyx 3000 series systems incorporate on-the-fly video resampling on all video channels. This feature is known as dynamic resolution, which works on all video formats with pixel rates of 120 MHz or less. This restriction eliminates only the highest-resolution formats.

This feature provides several benefits:

- Dynamic resolution, available through OpenGL Performer as well as OpenGL, provides a powerful method of guaranteeing scene update rates. Dynamic resolution works in addition to geometric level-of-detail methods.
- The video encoder channel can now encompass an entire video channel, allowing whole-screen recording without an external scan-converter. For video formats with pixel clock rates above 120 MHz, the encoder works in pass-through mode similar to RealityEngine2, allowing recording of a pannable NTSC- or PAL-size region of the display.
- Dynamic resolution of a frame buffer region allows a smaller frame buffer region to be zoomed or enlarged to fit the video format resolution of the video channel assigned to that frame buffer region. This aids efficient use of frame buffer memory without requiring nonstandard video formats.

7.14.8 Stereoscopic Video Output

SGI Onyx 3000 series systems continue the RealityEngine2 style of support for stereoscopic viewing in a window. Individual windows may be run in stereoscopic viewing mode, displaying a separate left-eye view and right-eye view at the appropriate frame rate. All other windows and pop-ups appear normal.

7.14.9 Direct Support of Color Field Sequential Video

SGI Onyx 3000 series systems provide direct, efficient support of time-multiplexed, sequential color field video, commonly known as color field-sequential video. Cursors and color index pop-up regions appear in their true colors, making it easy to integrate applications using color field-sequential video with the X-Windows environment.

7.14.10 Luminance

Luminance (monochrome rendering) is supported by the video display subsystem. Applications may select luminance to be displayed as color mapped pseudocolor or as static-gray X-visual. Ultrahigh color resolution (16 bits) is supported for rendering operations such as texture (but not lighting) using luminance.

7.14.11 Programmable Video Timing

The advanced circuitry of SGI Onyx 3000 series systems generates a programmable pixel clock to enable the broadest range of applications, from video production to visual simulation. The clock controls the rate at which the frame buffer is scanned out to a video signal. The standard system supports 1920x1200 pixels at 66 Hz noninterlaced.

Some applications need different video timing, such as military displays requiring the RS-343 standard, or stereoscopy for molecular modeling, which requires 120 Hz video field rate to output a left-eye view and a right-eye view at 60 Hz each. Programmable video timing in SGI Onyx 3000 series systems supports these and other video formats. With the Video Format Compiler, SGI Onyx 3000 series systems put video format programming capability in your hands.

Video formats produced by the Video Format Compiler must be put into a video format combination using the Video Format Combiner tool (Section 7.14.4) to validate that they will operate correctly on the SGI Onyx 3000 series video display subsystem hardware. In the case of a single format, the word combination is a misnomer; the key concept is that all video formats, whether running alone or in combination with other video formats, must be validated by the video combiner to ensure that the video format is compatible with the display hardware.

7.15 Genlock

To ensure that the SGI Onyx 3000 series InfiniteReality3 graphics subsystem video output is synchronized with other video sources or pipelines, the system includes a standard external genlock capability. Genlock ensures that multiple video sources all start scanning out their frames at the same time for frame-synchronized multichannel output.

Video channels in video subsystems of the SGI Onyx 3000 series are automatically frame-synchronized, regardless of whether they are of the same resolution and whether the external genlock feature is enabled. External genlock capability lets the master video format (the format running on the lowest numbered video channel) genlock to an external video source.

The external video source only needs the same vertical sync rate as the master format; it does not need the identical format running on the master channel. This provides maximum flexibility in synchronizing to multiple SGI Onyx 3000 series systems with InfiniteReality3 graphics pipelines or to video equipment of other vendors.

There are two methods of frame synchronization: genlock and frame-reset.

The highest-quality method of frame synchronization is genlock. It is used when two video formats are the same resolution and frame rate. The internal circuitry can make adjustments every horizontal period to ensure that the system is locked.

In the frame-reset method, the system merely issues a reset signal to the timing circuitry of the system to be synchronized. This method may allow frames to be dropped occasionally. Genlock guarantees that frames will not be dropped or missed.

The video combiner tool, IRCombine, will select the best-quality frame synchronizing method based on the video formats in the combination file and the video format selected for external genlocking.

7.15.1 Swap Synchronization

Swap buffer timing can be synchronized between any number of SGI Onyx 3000 series systems by means of external cabling to the swap_ready connector. The systems should be genlocked together when using swap synchronization.

7.16 Digital Video Multiplexer (DPLEX)

7.16.1 Breakthrough Scalability for Visualization

The Digital Video Multiplexer (DPLEX) option takes visualization scalability to a new level. DPLEX permits multiple InfiniteReality3 pipelines in an SGI Onyx 3000 series system to work simultaneously on a single suitable visual application. Moreover, DPLEX provides this capability in hardware, resulting in perfect (100 percent) scaling of both geometry rate and fill rate on DPLEX-capable applications. An n-pipe system equipped with DPLEX may support up to n times the frame rate of a single-pipe system running the same application, or n times the scene complexity for a given frame rate, provided the rest of the system and application scale.

DPLEX is an optional daughtercard for SGI Onyx 3000 series systems that enables digital multiplexing of two or more pipelines. One DPLEX option is required for each multiplexed pipeline in the system. The concept is simple: multiple pipelines operate simultaneously on successive frames of an application, which are then digitally multiplexed together before being converted to analog video. Any user seeking greater rendering speed may benefit from this technology. Key applications include distortion correction (required for dome simulators) and interactive large model visualization (required in many manufacturing settings, research institutions, etc.).

In addition to its multiplexing capabilities, DPLEX also provides high-resolution Low Voltage Differential Signal (LVDS) digital video output to drive next-generation digital displays, hardware-in-the-loop devices, and other custom peripherals.

7.16.2 Distortion Correction

With the SGI Onyx 3000 series, distortion correction is achieved using an approach in which a flat image is rendered to the frame buffer, copied back into texture memory, and then projected on a grid of polygons that represents an inverse mapping of the required distortion. A single InfiniteReality3 pipeline is capable of rendering 1280x1024 distortion-corrected imagery at 30 Hz. With DPLEX, two such pipes can be multiplexed together to deliver a 60 Hz update rate. And since the distortion correction occurs in the IG rather than in the projection system, the SGI Onyx 3000 series can support dynamic distortion correction, in which the relative location of the projector or the shape of the screen changes over time.

7.16.3 Large Model Interactivity

DPLEX may allow an application to achieve higher performance levels. This can be very useful in digital prototyping, where large models are required to maintain visual fidelity. With DPLEX, even the largest, most complex visualizations become manageable because performance can be scaled to meet virtually any challenge.

7.16.4 Application and OS Support

Full-screen applications written to OpenGL Performer or OpenGL Optimizer 1.1 are most easily adaptable to DPLEX operation. Other OpenGL applications will require additional modification to enable multipipe support in general and DPLEX support in particular. Not all applications can be made to work with DPLEX. DPLEX requires IRIX 6.5.2 or later.

7.16.5 DPLEX and MonsterMode Rendering

Underlying DPLEX is a data partitioning scheme known as time composition: multiple pipelines operate simultaneously on different time slices of a data set to generate successive frames, which are then composited together by the DPLEX network to form a continuous video stream. Because the composition takes place in dedicated hardware, DPLEX is said to perform hardware-based time composition.

The SGI Onyx 3000 series also supports several software-based methods for distributing a data set over multiple pipelines. These methods are collectively known as MonsterMode rendering. Each uses the high-bandwidth memory subsystem rather than special hardware (DPLEX) for the interpipeline transfers required to partition the data and compose the final image. MonsterMode rendering methods include 2D composition for handling polygonal models and 3D composition for handling volumetric models. The latter provides the further benefit of additive texture capacity: an n-pipe system using MonsterMode 3D-composition software has n times the effective texture memory of a single-pipe system (up to 4GB for a 16-pipe configuration). The choice of which method to use, DPLEX or MonsterMode, will depend on the nature of the application. Volume visualization should benefit more from MonsterMode, whereas polygonal data may favor DPLEX, MonsterMode, or a combination of the two. Please consult your SGI representative for more information.

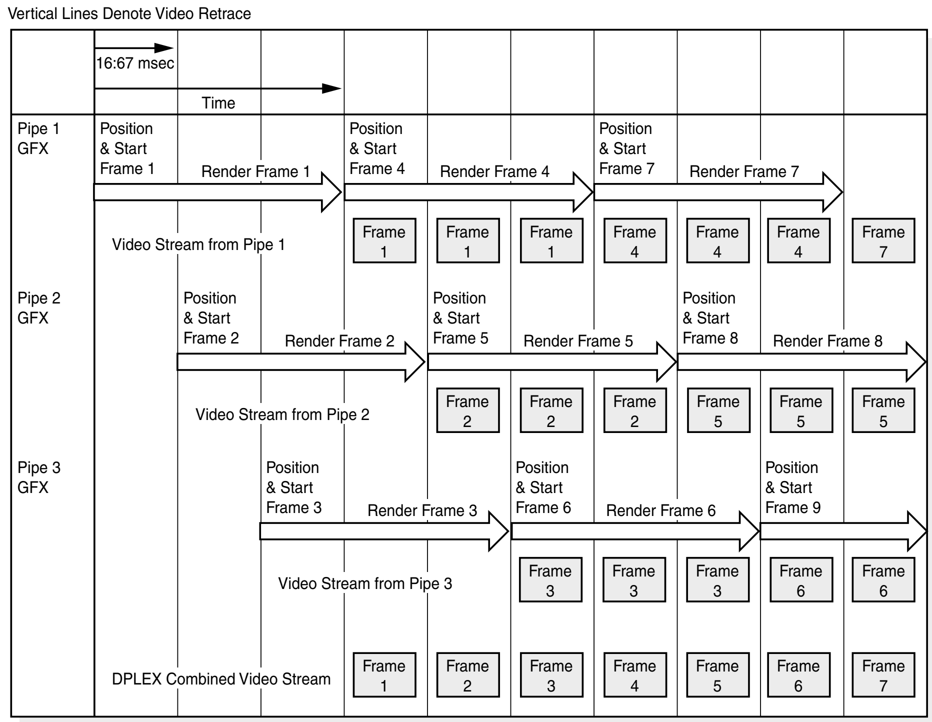


Figure 19. Example of 3-Pipe DPLEX Time Compositing

7.17 Integration with X Window System

SGI Onyx 3000 series systems are integrated with the X Window System, giving multiple, independent graphics displays from a single frame buffer. The X11 standard is fully supported and native OpenGL is integrated without degradation of graphics pipeline performance when drawing into windows (versus a dedicated full screen).

SGI Onyx 3000 series systems also unite a number of unique capabilities with respect to X Window System support. Up to 16 windows can have their own 12-bit color lookup tables (LUTs) to provide a stable, independent data space that does not require reloading with each current window switch.

7.18 Digital Media

A range of digital media options that allow SGI Onyx 3000 series systems to be connected to industry-standard video and audio equipment is available. Graphics-to-video output and memory input/output support is available for both HD and SD video. Standard audio input/output support provides the capability of precisely synchronizing numerous audio channels to a single video channel with available audio options allowing channel count to be increased to match application requirements.

7.18.1 Digital Video Option (DIVO)

The DIVO board is a half-size XIO board that is capable of streaming lossless uncompressed real-time SD video and audio to and from system memory. Each DIVO board includes two fully independent channels, one for input and one for output. Video output can be genlocked to external reference sources or operate standalone.

DIVO supports uncompressed serial digital video formats ranging from single-link 4:2:2 8-bit to dual-link 4:4:4:4 10-bit, specifically:

- SMPTE 259M (10-bit 4:2:2 Component Digital only)
- SMPTE 272M (video with embedded audio)
- SDDI
- NTSC and PAL timing
- Dual-link 4:4:4:4 RGBA or 4:2:2:4 YCrCbA I/O support

DIVO also supports Rice encode/decode of these uncompressed formats, providing lossless reduction of approximately 2:1, reducing disk storage and network bandwidth requirements. Additionally, DIVO supports up to 16 channels of embedded audio on both input and output and real-time, user-selectable, high-quality colorspace conversion during input and output.

7.18.2 Digital Video Option—DVCPRO (DIVO-DVC)

The DIVO-DVC provides all of the functionality of the DIVO board plus real-time DVCPRO 25 encoding and decoding for the input and output channels.

7.18.3 Graphics-to-Video Out (GVO) Option

The GVO option is a daughtercard that connects directly to the graphics pipeline of SGI Onyx 3000 series systems and allows the user to output CCIR 601 serial digital video from the graphics frame buffer. Video is transferred from the graphics pipeline to external devices in real time. 720x486 (NTSC) or 720x576 (PAL) video window sizes are supported plus a larger screen area can be rescaled to one of these sizes.

GVO supports uncompressed serial digital video formats ranging from single-link 4:2:2 8-bit to dual link 4:4:4:4 10-bit, specifically:

- SMPTE 259M (10-bit 4:2:2 Component Digital only)
- SMPTE 272M (video with embedded audio)
- SDDI

- NTSC and PAL timing
- 4:4:4:4 (dual-link) RGBA or 4:2:2:4 (dual-link) YCrCbA (8- or 10-bit)

The GVO also supports real-time, user-selectable, high-quality colorspace conversion during input and output. Connections to the GVO are serial digital.

When the GVO card is installed on a Display Generator with two channels, both channels remain available. When installed on a Display Generator with eight channels, three channels remain available.

7.18.4 High-Definition Digital Video (HD I/O) Option

The HD I/O option is an XIO board that is capable of delivering uncompressed real-time HD video to and from system memory. Each HD I/O board includes two fully independent channels, one for input and one for output. Video output can be genlocked to external reference sources or operate standalone.

HD I/O supports a wide range of uncompressed HD digital video formats, including:

- SMPTE 274M (1080i); 1920x1080i @ 59.94 Hz
- SMPTE 296M (720p); 1280x720p @ 59.94 Hz
- 1080 24p, 24psF, 25p and 25 psF
- 1035I @ 59.94 Hz
- 4:2:2 YCrCb sampling with 8 or 10 bits per component
- Full sample rate alpha channel (8- or 10-bit)
- 4:4:4:4 (dual-link) RGBA or 4:2:2:4 (dual-link) YCrCbA (8- or 10-bit)

The HD I/O also supports real-time, user-selectable, high-quality colorspace conversion during input and output. Connections to the HD I/O are parallel digital; external serializer/deserializers enable the connection of equipment requiring serial connections.

7.18.5 High-Definition Graphics-to-Video Out (HD GVO) Option

The HD GVO option is a daughtercard that connects directly to the graphics pipeline of SGI Onyx 3000 series systems and allows the user to output uncompressed real-time HD or SD digital video formats from the graphics frame buffer using the Video BreakOut Box (VBOB).

A wide variety of uncompressed video formats are supported, including:

- 480i and 576i SD formats
- SMPTE 274M (1080i); 1920x1080i @ 59.94 Hz
- SMPTE 296M (720p); 1280x720p @ 59.94 Hz
- 4:2:2:4 YCrCbA sampling with 8 or 10 bits per component

The HD GVO also supports real-time, user-selectable, high-quality colorspace conversion. Connections to the HD GVO are serial digital for HD (SMPTE 292M) and SD (SMPTE 259M).

System flexibility allows the addition of future video formats and frame rates via software upgrades.

7.18.6 DMediaPro™ DM2

The DMediaPro DM2 option is capable of delivering uncompressed real-time HD and SD video to and from system memory. Using VBOB, DMediaPro DM2 includes two fully independent channels, one for input and one for output. Video output can be genlocked to external reference sources or input video or operate standalone.

DMediaPro DM2 supports uncompressed SD serial digital video formats, including:

- SMPTE 259M (10-bit 4:2:2 component digital only)
- NTSC and PAL timing
- Single-link 4:2:2 YCrCb I/O support
- Dual-link 4:4:4:4 RGBA or 4:2:2:4 YCrCbA I/O support

DMediaPro DM2 supports a wide range of uncompressed HD serial digital video formats including:

- SMPTE 274M (1080i); 1920 x 1080i @ 50, 59.94 and 60 Hz
- SMPTE 296M (720p); 1280 x 720p @ 59.94 and 60 Hz
- 1080p; 1920 x 1080p @ 23.976, 24, and 25 Hz
- 1080PsF; 1920 x 1080PsF @ 23.976, 24 and 25PsF
- 4:2:2 YCrCb or 4:4:4 RGB sampling with 8 or 10 bits per component
- Dual-link 4:2:2:4 YCrCbA or 4:4:4:4 RGBA sampling with 8 or 10 bits per component

User-selectable, real-time, high-quality colorspace conversion is supported during input and output. VBOB provides serial digital video connections for HD and SD video I/O and analog connections for genlock.

7.18.7 Digital Audio

The Digital Audio system for the SGI Onyx 3000 series is a half-size PCI board that provides up to 10 channels of digital audio. Additional Digital Audio boards can be installed to expand the audio capabilities of the SGI Onyx 3000 series.

A full complement of audio features is supported, including:

- Digital audio input
  - AES3-75 ohm BNC connector
  - Can be used as an AES11 synchronization input (audio clock rates)
  - AES3 professional 2-channel, 24-bit
  - IEC958, S/PDIF consumer 2-channel digital
- Digital audio output
  - AES3-75 ohm BNC connector
  - AES3 professional 2-channel, 24-bit
  - IEC958, S/PDIF consumer 2-channel digital

- ADAT Optical digital audio input
  - 12.8MB/second SHARP multimode plastic fiber-optic connector
  - Can be used as a synchronization input
  - 8-channel, 24-bit
  - 2-channel consumer digital (IEC958, S/PDIF format) via optical
- ADAT Optical digital audio output
  - 12.8MB/second SHARP multimode plastic fiber-optic connector
  - 8-channel, 24-bit
  - 2-channel consumer digital (IEC958, S/PDIF format) via optical
- Video reference input for synchronization

The Digital Audio system's independent inputs and outputs can be used simultaneously either with independent or synchronized sample rates. In addition, sample-accurate timing information enables precise synchronization.