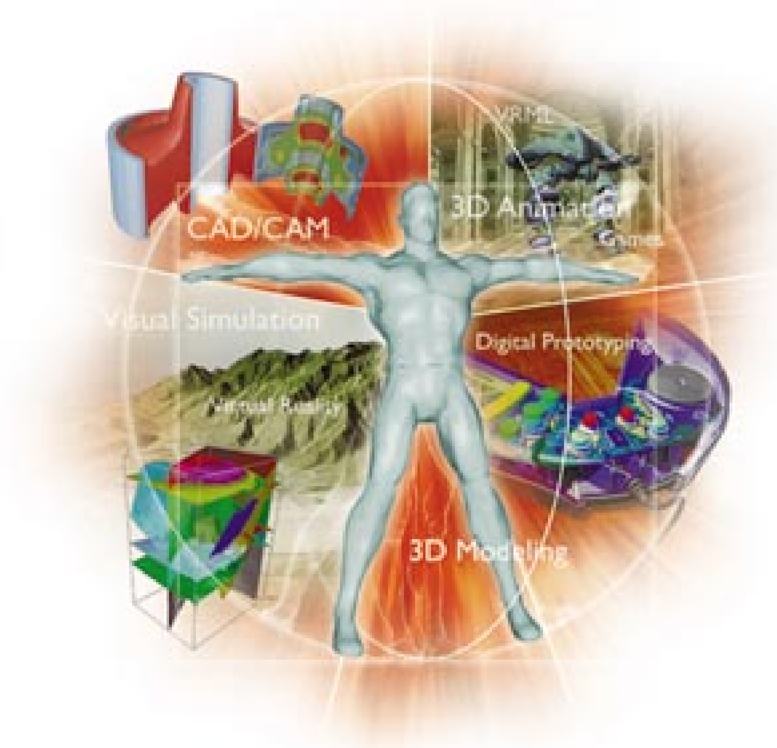
OpenGL®

The Industry's Foundation for High-Performance Graphics



Most Widely Adopted Graphics Standard

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

High Visual Quality and Performance

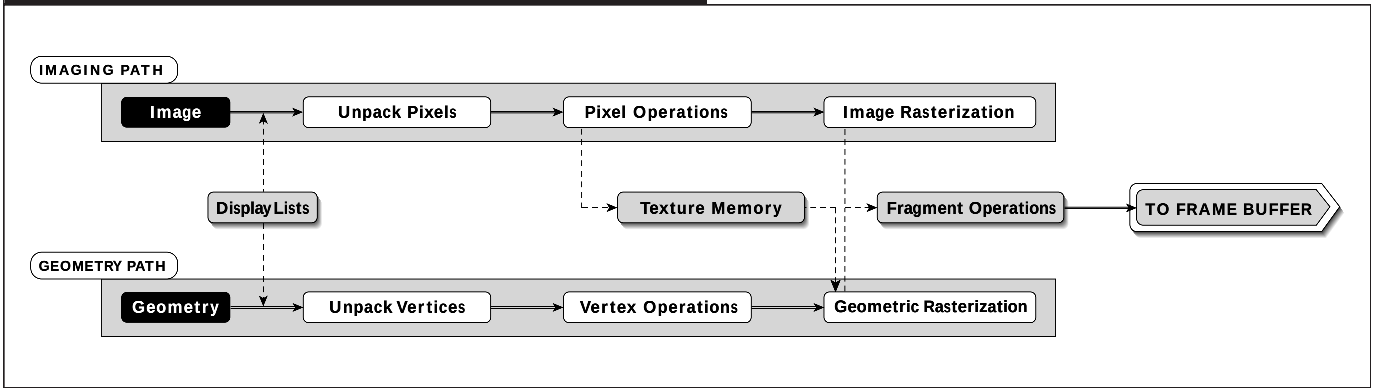
Any visual computing application requiring maximum performance—from 3D animation to CAD to visual simulation—can exploit high-quality, high-performance OpenGL capabilities. These capabilities allow developers in diverse markets such as broadcasting, CAD/CAM/CAE, entertainment, medical imaging, and virtual reality to produce and display incredibly compelling 2D and 3D graphics.

Developer-Driven Advantages

- **Industry standard.** An independent consortium, the OpenGL Architecture Review Board, guides the OpenGL specification. With broad industry support, OpenGL is the only truly open, vendor-neutral, multiplatform graphics standard.
- **Stable.** OpenGL implementations have been available for more than seven years on a wide variety of platforms. Additions to the specification are well controlled, and proposed updates are announced in time for developers to adopt changes. Backward compatibility requirements ensure that existing applications do not become obsolete.
- **Reliable and portable.** All OpenGL applications produce consistent visual display results on any OpenGL API-compliant hardware, regardless of operating system or windowing system.
- **Evolving.** Because of its thorough and forward-looking design, OpenGL allows new hardware innovations to be accessible through the API via the OpenGL extension mechanism. In this way, innovations appear in the API in a timely fashion, letting application developers and hardware vendors incorporate new features into their normal product release cycles.
- **Scalable.** OpenGL API-based applications can run on systems ranging from consumer electronics to PCs, workstations, and supercomputers. As a result, applications can scale to any class of machine that the developer chooses to target.
- **Easy to use.** OpenGL is well structured with an intuitive design and logical commands. Efficient OpenGL routines typically result in applications with fewer lines of code than those that make up programs generated using other graphics libraries or packages. In addition, OpenGL drivers encapsulate information about the underlying hardware, freeing the application developer from having to design for specific hardware features.
- **Well-documented.** Numerous books have been published about OpenGL, and a great deal of sample code is readily available, making information about OpenGL inexpensive and easy to obtain.

Datasheet

The OpenGL Visualization Programming Pipeline



OpenGL operates on image data as well as geometric primitives.

Simplifies Software Development, Speeds Time-to-Market

OpenGL routines simplify the development of graphics software—from rendering a simple geometric point, line, or filled polygon to the creation of the most complex lighted and texture-mapped NURBS curved surface. OpenGL gives software developers access to geometric and image primitives, display lists, modeling transformations, lighting and texturing, anti-aliasing, blending, and many other features.

Every conforming OpenGL implementation includes the full complement of OpenGL functions. The well-specified OpenGL standard has language bindings for C, C++, Fortran, Ada, and Java™. All licensed OpenGL implementations come from a single specification and language binding document and are required to pass a set of conformance tests. Applications utilizing OpenGL functions are easily portable across a wide array of platforms for maximized programmer productivity and shorter time-to-market.

All elements of the OpenGL state—even the contents of the texture memory and the frame buffer—can be obtained by an OpenGL application. OpenGL also supports visualization applications with 2D images treated as types of primitives that can be manipulated just like 3D geometric objects. As shown in the OpenGL visualization programming pipeline diagram above, images and vertices defining geometric primitives are passed through the OpenGL pipeline to the frame buffer.

Available Everywhere

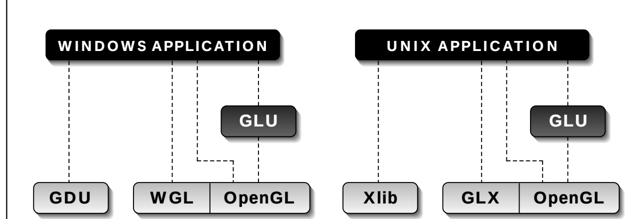
Supported on all UNIX® workstations, and shipped standard with every Windows NT® and Windows® 95 PC, no other graphics API operates on a wider range of hardware platforms and software environments. OpenGL runs on every major operating system including Mac® OS, OS/2®, UNIX, Windows 95, Windows NT, Linux, OPENStep, Python, and BeOS; it also works with every major windowing system, including Presentation Manager, Win32, and X/Window System™. OpenGL is callable from Ada, C, C++, Fortran, and Java and offers complete independence from network protocols and topologies.

Architected for Flexibility and Differentiation

Although the OpenGL specification defines a particular graphics processing pipeline, platform vendors have the freedom to tailor a particular OpenGL implementation to meet unique system cost and performance objectives. Individual calls can be executed on dedicated hardware, run as software routines on the standard system CPU, or implemented as a combination of both dedicated hardware and software routines. This implementation flexibility means that OpenGL hardware acceleration can range from simple rendering to full geometry and is widely available on everything from low-cost PCs to high-end workstations and supercomputers. Application developers are assured consistent display results regardless of the platform implementation of the OpenGL environment.

Using the OpenGL extension mechanism, hardware developers can differentiate their products by developing extensions that allow software developers to access additional performance and technological innovations.

API Hierarchy



- **OpenGL applications use the window system's window, input, and event mechanism**
- **GLU supports quadrics, NURBS, complex polygons, matrix utilities, and more**

This diagram demonstrates the relationship between OpenGL GLU and windowing APIs.

The Foundation for Advanced APIs

Leading software developers use OpenGL, with its robust rendering libraries, as the 2D/3D graphics foundation for higher-level APIs. Developers leverage the capabilities of OpenGL to deliver highly differentiated, yet widely supported vertical market solutions. For example, Open Inventor™ provides a cross-platform user interface and flexible scene graph that makes it easy to create OpenGL applications. IRIS Performer™ leverages OpenGL functionality and delivers additional features tailored for the demanding high frame rate markets such as visual simulation and virtual sets. OpenGL Optimizer™ is a toolkit for real-time interaction, modification, and rendering of complex surface-based models such as those found in CAD/CAM and special effects creation. The Fahrenheit Scene Graph, scheduled for release in 1999, will leverage OpenGL capabilities to provide a platform for applications and APIs across diverse market segments, allowing reduced development time, maximized performance, and high visual quality.

Governance

The OpenGL Architecture Review Board (ARB), an independent consortium formed in 1992, governs the OpenGL specification. Composed of members from many of the industry's leading graphics vendors, the ARB defines conformance tests and approves OpenGL enhancements. Currently the board includes representatives from Compaq, Evans & Sutherland, Hewlett-Packard, IBM, Intel, Intergraph, Microsoft, and Silicon Graphics. The OpenGL ARB Web site can be found at *www.opengl.org.*

The OpenGL Performance Characterization Committee, another independent organization, creates and maintains OpenGL benchmarks and publishes the results of those benchmarks on its Web site: *www.specbench.org/gpc/opc.static/index.html.*

Continued Innovation

The OpenGL standard is constantly evolving. Formal revisions occur at periodic intervals, and extensions allowing application developers to access the latest hardware advances through OpenGL are continuously being developed. As extensions become widely accepted, they are considered for inclusion into the core OpenGL standard. This process allows OpenGL to evolve in a controlled yet innovative manner.

In the most recent revision of OpenGL (version 1.2), several capabilities that were previously available as extensions were rolled into the core OpenGL standard. (See back page for more details.)

Licensing

ARB-approved OpenGL specifications and source code are available to licensed hardware platform vendors. End users, independent software vendors, and others writing code based on the OpenGL API are free from licensing requirements. See *www.opengl.org/Developers/developers.html* for more information.

OpenGL Applications *(partial list)*

Company	Application
3D ANIMATION AND MODELING	
Alias|Wavefront	Maya™
Byte by Byte	Soft F/X 3D
Caligari	truSpace
CrystalGraphics	Crystal 3D Impact Pro
Hash Inc.	Hash 3D Animation System
Hollywood FX Inc.	Hollywood FX
Kinetix	3D Studio Max, Character Studio
Lightscape	Lightscape
MultiGen	GameGen™, MultiGen® Creator
Newtek	LightWave 3D
Nichimen Graphics	N-World
Softimage	Digital Studio, Softimage 3D®
Strata	StudioPro™
Template Graphics Software	LiveWork 3D, 3Space Publisher
CAD/CAM & DIGITAL PROTOTYPING	
Bentley Systems	MicroStation®
CoCreate	SolidDesigner
Dassault Systèmes	CATIA®
EDS Unigraphics	Unigraphics®
Kinetix	3D Studio VIZ
Matra Datavision	EUCLID Quantum
Parametric Technology Corp.	Pro/ENGINEER®, PT/Modeler
Parasolid	UG/Creator
SDRC	I-DEAS™
SolidWorks	SolidWorks
VISUAL SIMULATION AND VIRTUAL REALITY	
Advanced Visual Systems	AVS/Express™
Coryphaeus Software	Designer's Workbench™
DataPath	RealiMation
IBM	Visualization Data Explorer
MultiGen	MultiGen II Pro
Paradigm Simulation	Vega
Research Systems	IDL®
Sense8/Engineering Animation Inc.	WorldToolKit™
Silicon Graphics	IRIS Performer
VRML	
Cosmo Software	Cosmo™ World, PageFX
Ligos Technology	V-Realm Builder
Rendersoft	Rendersoft VRML Editor
Sense8/Engineering Animation Inc.	World Up
Systems in Motion	PolyRed
GAMES	
Epic Megagames	Unreal
id Software	Quake2
ION Storm	Dai-Katana, Anachronox
Laminar Research	X-Plane
Ritual Entertainment	SiN
Valve Software	Half-Life
Zombie VR Studios	SpecOps

OpenGL

OPENGL GRAPHICS FUNCTIONS *(partial list)*

Accumulation buffer. A buffer in which multiple rendered frames can be composited to produce a single blended image. Used for effects such as depth of field, motion blur, and full-scene anti-aliasing.

Alpha blending. Provides a means to create transparent objects. Using alpha information, an object can be defined as anything from totally transparent to totally opaque.

Anti-aliasing. A rendering method used to smooth lines and curves. This technique averages the color of the pixels adjacent to the line. It has the visual effect of softening the transition of the pixels on the line and those adjacent to the line, thus providing a smoother appearance.

Color-index mode. Color buffers store color indices rather than red, green, blue, and alpha color components.

Display list. A named list of OpenGL commands. The contents of a display list may be preprocessed and might therefore execute more efficiently than the same set of OpenGL commands executed in immediate mode.

Double buffering. Used to provide smooth animation of objects. Each successive scene of an object in motion can be constructed in the back or "hidden" buffer and then displayed. This allows only complete images to ever be displayed on the screen.

Feedback. A mode where OpenGL will return the processed geometric information (colors, pixel positions, and so on) to the application as compared to rendering them into the frame buffer.

Gouraud shading. Smooth interpolation of colors across a polygon or line segment. Colors are assigned at vertices and linearly interpolated across the primitive to produce a relatively smooth variation in color.

Immediate mode. Execution of OpenGL commands when they're called, rather than from a display list.

Materials lighting and shading. The ability to accurately compute the color of any point given the material properties for the surface.

Pixel operations. Storing, transforming, mapping, zooming.

Polynomial evaluators. To support non-uniform rational B-splines (NURBS).

Primitives. A point, line, polygon, bitmap, or image. Raster primitives: Bitmaps and pixel rectangles.

RGBA mode. Color buffers store red, green, blue, and alpha color components, rather than indices.

Selection and picking. A mode in which OpenGL determines whether certain user-identified graphics primitives are rendered into a region of interest in the frame buffer.

Stencil planes. A buffer that can be used to mask individual pixels in the color frame buffer.

Texture mapping. The process of applying an image to a graphics primitive. This technique is used to generate realism in images. For example, a tabletop drawn as a rectangle could have a wood-grain texture applied to it to make it look more realistic.

Transformation. The ability to change the rotation, size, and perspective of an object in 3D coordinate space.

Z-buffering. The Z-buffer is used to keep track of whether one part of an object is closer to the viewer than another. It is important in hidden surface removal.

NEW CORE CAPABILITIES OF OPENGL 1.2

The latest specification of the OpenGL API defines new core capabilities for all 1.2 implementations and a new optional imaging subset.

- *Three-dimensional texturing* for supporting hardware-accelerated volume rendering
- *BGRA pixel formats* and *packed pixel formats* to directly support more external file and hardware frame buffer types
- *Automatic rescaling of vertex normals* changed by the modeling matrix; rescaling can, in some cases, replace a more expensive renormalization operation
- Application of *specular highlights after texturing* for more realistic lighting effects
- *Texture coordinate edge clamping* to avoid blending border and image texels during texturing
- *Level of detail control* for mipmap textures to allow loading only a subset of levels; this can save texture memory when high-resolution texture images are not required due to textured objects being far from the viewer
- *Vertex array enhancements* to specify a subrange of the array and draw geometry from that subrange in one operation; this allows a variety of optimizations such as pretransforming, caching transformed geometry, and so on

OPTIONAL IMAGING FEATURES OF OPENGL 1.2

The optional imaging subset includes a variety of enhancements to the OpenGL pixel path. An implementation may or may not support the subset, but any implementation that supports it must provide all of the documented features.

The imaging subset includes all of the following features:

- Blending enhancements include specifying a *constant color* as one component of a blend as well as a way to define new blending equations; some of the new equations supported include the ability to *choose the minimum or maximum of the blend colors* and to *subtract one of the blend colors* from the other
- A *color matrix stage*, which transforms colors under an arbitrary 4x4 matrix, may be enabled after convolution; it may be used to reassign and duplicate color components or for simple color space conversions
- *Color tables* are an RGBA format color look-up mechanism capable of operating on a subset of color components; it is possible to respecify parts of a color table without recreating the table
- Simple *1D and 2D convolutions* may then be done; both separable and nonseparable 2D filters are allowed; new convolution border modes define several possible behaviors when the filter samples both image and border pixels
- The pixel path may gather *histogram statistics* that the application may query

SiliconGraphics
Computer Systems

Corporate Office
2011 N. Shoreline Boulevard
Mountain View, CA 94043
(650) 960-1980
URL: http://www.sgi.com

U.S. 1(800) 800-7441
Europe (44) 118-925.75.00
Asia Pacific (81) 3-54.88.18.11
Latin America 1(650) 933.46.37

Canada 1(905) 625-4747
Australia/New Zealand (61) 2.9879.95.00
SAARC/India (91) 11.621.13.55
Sub-Saharan Africa (27) 11.884.41.47

See *www.opengl.org* for daily OpenGL updates.

434 (7/98)