Datasheet

OpenGL Shader™

Features

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

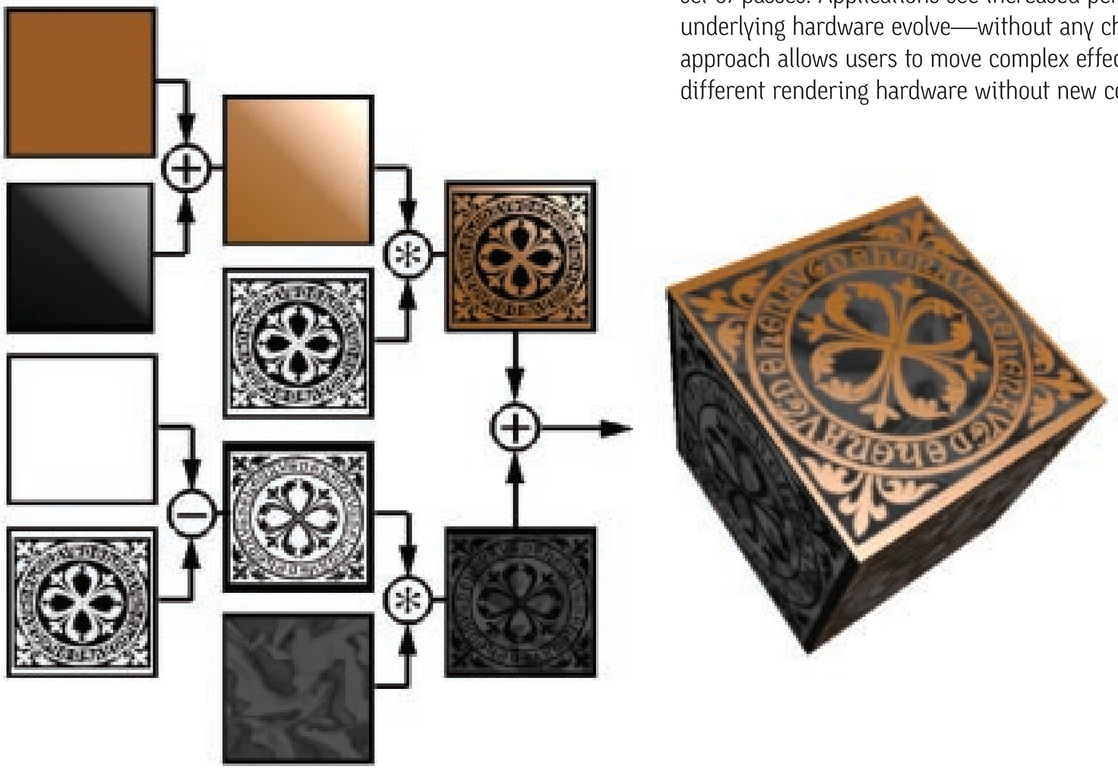
Is It Real, or Is It Rendered?

Movie special effects deliver stunning realism, but cannot be directly rendered on today's graphics systems. These effects are designed using shaders—high-level descriptions of complex appearances—which are then rendered over many hours on powerful computers, a process known as offline rendering. With the advent of powerful OpenGL® graphics pipelines such as SGI™ InfiniteReality3™, many programmers have developed complex multipass algorithms that mimic these effects.

OpenGL Shader enables these powerful shaders to be directly rendered by OpenGL systems much more rapidly, transforming the reach and usability of these powerful descriptions. Every aspect of the scene is encapsulated in these descriptions, such as material properties, fog, and other atmospherics. Eliminating the synthetic feel, anisotropic and full bidirectional reflectivity is an integral component of this natural environment.

See Hardware Improvements without Software Recompiles

OpenGL Shader transforms programmable shading from an offline tool to a central component of interactive visual computing. OpenGL Shader accomplishes this transformation by introducing a compiler between the application and the graphics library that translates shaders into OpenGL rendering passes. The compiler can produce a general set of rendering passes or use knowledge of the available hardware to select an optimized set of passes. Applications see increased performance as the compiler and underlying hardware evolve—without any change in the shaders. This approach allows users to move complex effects they have developed to different rendering hardware without new coding effort.



Everything You Need in One Development Kit

OpenGL Shader is a software development kit containing tools for supporting interactive, programmable shading on OpenGL systems. It consists of command line compilers and translators that can convert a set of Interactive Shading Language™ shaders into an OpenGL function call, as well as an Interactive Shading Language Library that enables applications to access the compilers in an interactive system. Documentation and sample source code are also included in the OpenGL Shader software development kit. The source code examples include a stand-alone application and an Open Inventor™ toolkit-based application.

Interactive Shading Language Library

The OpenGL Shader Interactive Shading Language Library provides a minimal interface for supporting interactive, programmable shading. The Interactive Shading Language Library consists of six classes that enable an application to:

- Define an appearance consisting of Interactive Shading Language shaders
- Compile that appearance into an OpenGL stream
- Associate it with geometry from the application
- Render the shaded geometry to an OpenGL rendering context by the application

Command Line Compiler

The command line compiler translates an appearance description into a description of OpenGL passes. When converted to an OpenGL stream using a translator, this pass description will render an object with the specified appearance. An appearance is defined as one or more of the following: a list of surface shaders, a list of ambient light shaders, and a list of direct light shaders. The shaders are written in an OpenGL Interactive Shading Language.

Command Line Translator

The command line translator translates a description of OpenGL passes, as output by the command line compiler, into C code that implements the OpenGL passes described in the input. For a given intermediate pass file, one .c file and one .h file are generated. The .c file contains the definitions of the initialization, drawing, and cleanup functions for the shader, while the .h file contains the prototypes for these functions.

For more information or to download a copy, check out *www.sgi.com/software/shader*.

OpenGL Shader Version 2.0
Feature Summary

- Support of enhanced Interactive Shading Language:
  - Enables general texture coordinate computation
  - Enables bump mapping within shaders
  - Enables general bidirectional reflectivity distribution functions within shaders

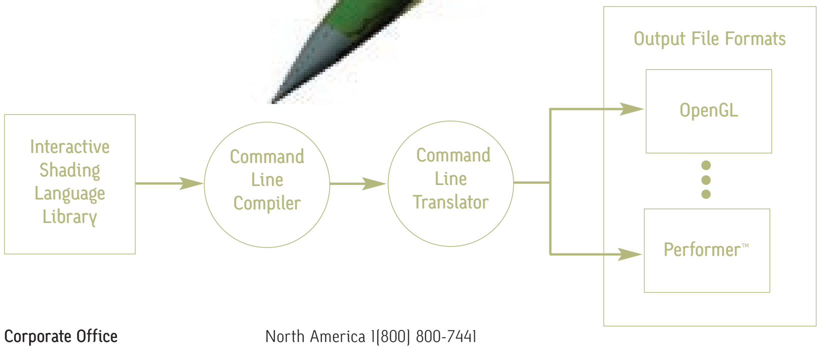
- Interactive Shading Language command line compiler to intermediate pass file:
  - Optimized code generation beyond Version 1.0

- Command line translator from intermediate pass file to OpenGL source code that can be compiled directly into an application

- Compiler library and OpenGL interpreter:
  - Can invoke Interactive Shading Language compiler from within any application
  - Simple API ensures minimal invasiveness
  - Supports changing shader parameters on the fly
  - Supports interactive shader design

- Example code:
  - Example application to display generated OpenGL source code
  - Sample shaders
  - Example of plugging library into a retained mode interface
  - Example shader editor

OpenGL Shader Architecture



sgi™

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

North America 1(800) 800-7441
Latin America 1(650) 933-4637
Europe (44) 118.925.75.00
Japan (81) 3.5488.1811
Asia Pacific (65) 771.0290

2954 (11/00)

J12018