SDK White Paper
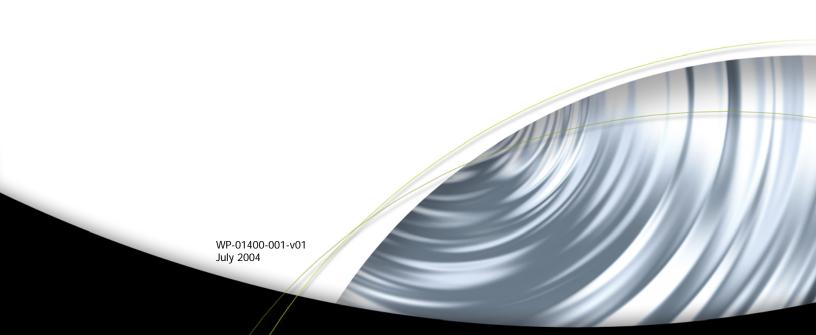
Vertex Lighting
Achieving fast lighting results

nVSDK

Vertex Lighting

Vertex lighting is not as accurate as per-pixel lighting. However, it has the advantage of being very fast. The lighting calculations are done only per vertex. This document describes briefly several implementations of vertex lighting, namely:

❑ Point lights with specular characteristics

❑ Directional lights with specular characteristics

❑ Two-sided lighting

❑ 17 simple-point diffuse-only light implementation.

These sample implementations do not use any pixel shader, in the interest of focusing on the vertex shader. However, the vertex shader can be combined with more complex pixel shader operations to achieve great results.

All techniques are implemented under DirectX9.0b and run on VS1.1.

Bryan Dudash
bdudash@nvidia.com

NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
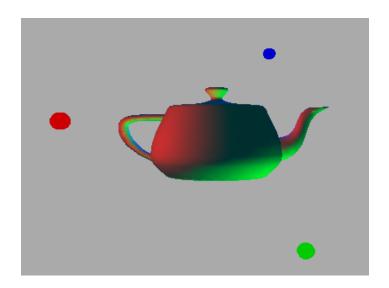
Diffuse Specular Point
(3 lights)

To use diffuse specular point lighting, calculate the dynamic vertex color using the standard Blinn diffuse and specular equations one time per light:

$$Diffuse = DiffuseColor * (N \bullet L)$$

$$Specular = SpecularColor * (H \bullet N)^{spec}$$

$$Attenuation = A_0 + A_1 d + A_2 d^2$$

The resulting color from each light is added into a final result per vertex. The diffuse color and specular color inputs, as well as the light direction, are supplied as constants for each light source.

Note: For point lights, the specular and diffuse components are attenuated based on distance(*d*).
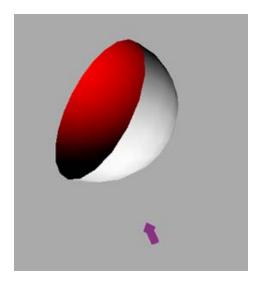
Diffuse Specular Directional (3 lights)

Diffuse specular directional lighting equations are done once per light and the resultant color from each light is added into a final result per vertex.

The lighting calculations for these are the same as the diffuse specular point lights described in the previous section, except there is no attenuation involved and the light direction is passed down to the shader instead of the light position.
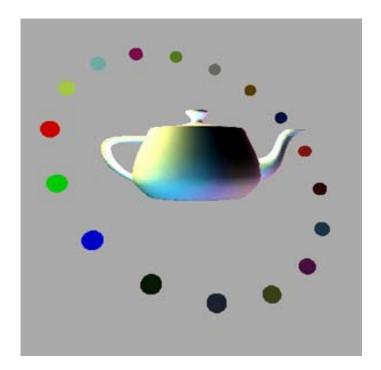
Two-sided Diffuse (1 light)

Two-sided diffuse lighting applies lighting to both sides of a model. A face normal is passed down as an extra piece of information to the vertex shader in the vertex stream. As a result this technique cannot be used with indexed vertex buffers.

Many Point (17 lights)

A many-point lighting technique implements a simple diffuse lighting equation based on vertex position and light position for 17 lights. This technique demonstrates how to implement an extremely cheap light in hardware and to obtain many light sources.

It compiles to PS1.1.

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2004 NVIDIA Corporation. All rights reserved