Technical Brief

Perfect Reflections and Specular Lighting Effects With Cube Environment Mapping

NVIDIA

Perfect Reflections and Specular Lighting Effects With Cube Environment Mapping

NVIDIA's introduction of cube environment mapping in the GeForce 256™ GPU (Graphics Processing Unit) is a 3D graphics breakthrough that enables developers to incorporate accurate, real-time reflections into the latest 3D games and applications. A unique feature of the NVIDIA GeForce 256, cube environment mapping is fully supported by Microsoft® DirectX® 7 and OpenGL® and is an easy-to-implement step toward photorealism on a desktop PC. Previous attempts to create reflections cheaply introduced severe limitations such as image warping and viewpoint dependency, and proved too costly in human effort or computer time. Through the revolutionary GeForce 256, the first chip to accelerate cube environment mapping in hardware, NVIDIA delivers a powerful tool for simulating complex, realistic reflections and specular lighting effects easily and cheaply.

Introduction

Imagine nature without reflections. While the goal of the PC graphics industry is to perfect the realism of the visual experience, certain rendering tasks such as accurate, real-time reflections have proven too difficult to implement practically. It often only takes a single artifact or incorrect surface detail to make a computed scene look fake and objectionable. Creating photorealistic "virtual environments" requires that surfaces are capable of reflecting light as they would in the natural world. Yet using today's hardware and common software tricks, it is impossible to create accurate reflections without difficult software implementations, objectionable artifacts, and disastrous performance hits. As a result, computer-synthesized environments are often simple and less interesting as developers have been forced to avoid reflective surfaces.

With GeForce 256's cube environment mapping, mainstream PC users will be able to experience stunning real-time, accurate reflections and specular lighting effects that are easy for developers to create. With cube environment mapping, developers will be able to utilize correct reflections to enrich 3D scenes and enhance game play. Accurate, real-time reflections will allow characters to gain insight into the surrounding 3D scenes, perhaps revealing the next vital step in game play. Cube environment mapping offers a quantum leap in visual realism that is easy to implement, fully supported by DirectX 7 and OpenGL, and which is very low-cost in computational resources.

Environment Mapping: Creating Reflections Efficiently

The surfaces of real-life objects are characterized by the degree to which light is absorbed, reflected or transmitted. In order to replicate the visual richness of natural objects on a mainstream PC, the eye must perceive that light is accurately reflected off objects in real-time without objectionable artifacts. The complexity involved in modelling the physical behavior of light by explicitly tracing secondary light rays throughout a scene, however, has led to alternative techniques to simulate easily realistic reflections.

Environment mapping is an efficient technique to compute reflections in computer-synthesized environments without tracing a myriad of secondary rays. By using pre-defined images cast as a geometric map shape that surrounds an object, reflections may be defined for a single point in space. Environment mapping samples a texture image from the environment map based on the reflected site vector for each point on the object (*Figure 1*). By projecting the 3D environment onto a 2D environment map that surrounds an object, reflections can be created with some degree of accuracy. Environment

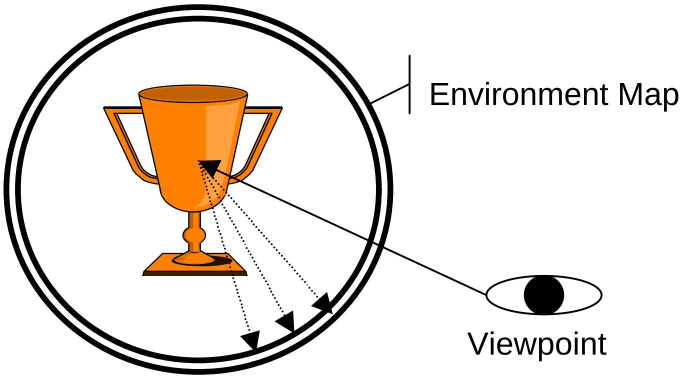

Figure 1: Environment Mapping For *each reflected "secondary ray" off a point on an object, a look-up in the surrounding environment map is done to provide a "reflected" image.*

mapping allows the number of computations needed to simulate reflections to be greatly reduced while producing results that maintain a high degree of image quality.

The Limits of Current Environment Mapping Techniques: Sphere Mapping

While many graphics chips may claim to support environment mapping as a major image quality feature, environment mapping is actually rarely used by designers due to severe limitations imposed by the *map shape*. Environment mapping, as it has been commonly defined, uses a sphere as a map shape in which a reflection is sampled from the map as a single spherically warped image (*Figure 2*). Unfortunately for developers, sphere mapping is useful only for one orientation and one viewpoint, limiting its use to very specific conditions that can not be violated. While sphere mapping can produce satisfactory reflections under exactly the right conditions, practically the combined limitations of *image warping and distortion, viewpoint dependency, the inability to render real-time, and the costly human effort and computational time required* to implement the sphere technique have meant the near-obsolescence of sphere mapping as an accessible feature.

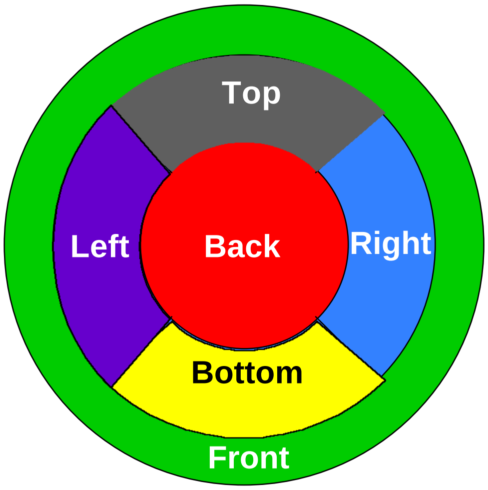

Figure 2: Spherical Environment Map Shape *With a spherical shape, the outer ring represents a single point in space, resulting in a stretched and warped image.*

Image Warping and Singularities With Sphere Mapping

Imagine the difficulty of taking a rectangular photograph and attempting to map it thoroughly to the inside of a sphere. When a texture is mapped onto the surface of a sphere it must be stretched and compressed. Even using "pre-stretching" and other tricks, warped images and "singularities" can not be avoided. Warping and distortion is a natural consequence of sphere mapping, particularly along the entire edge of the sphere which represents a single point in space. Singularities are mathematical discontinuities that occur with sphere mapping because the point behind the object is represented by the entire outer ring of the spherical environment map, which makes rendering a reflection on the back of an object impossible (*Figures 3 and 4*). Even if it were possible to change the viewpoint, with the sphere mapping technique an image rendered on the back of an object would be incorrect because the sampling against the stretched environment map will produce incorrect coordinates.

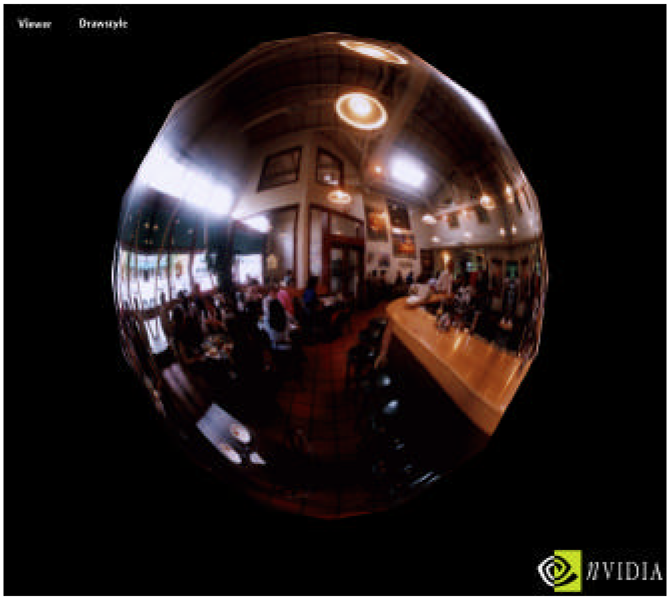

Figure 3: Sphere Environment Mapped Bubble
The reflection off the bubble above was created using the sphere mapping technique.

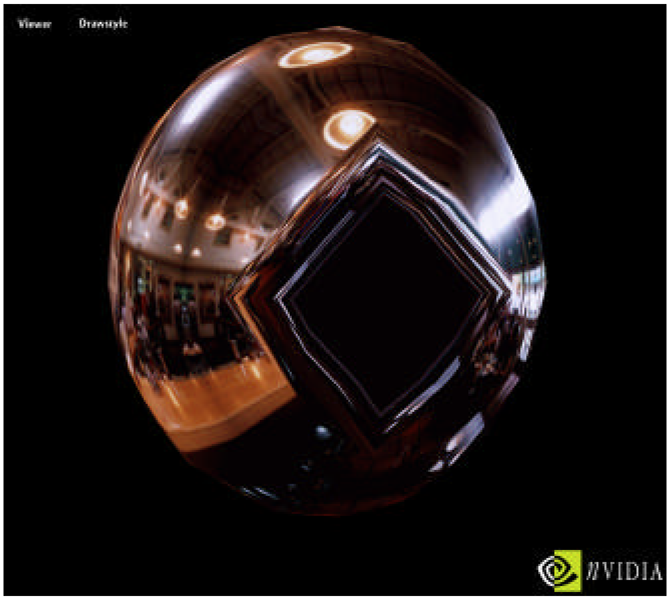

Figure 4: The Back of Bubble
Assuming sphere mapping allowed for viewpoint independence, this is the backside of the same sphere mapped reflection. The map shape unavoidably produces incorrect coordinates resulting in a damaged image.

The sphere mapping technique forces developers to exercise an incredible amount of time and forethought just to get a composite that isn't "too objectionable." Designers are concluding, however, that this is not time well spent since sphere mapping produces a satisfactory reflection for only one orientation and view.

Viewpoint Dependency and Real-time Drawing Limitations

If a character moves in a scene, the viewpoint will also change. The heart of the problem facing sphere mapping is that there is no straightforward camera model for real-time drawing. This is an unacceptable restriction of sphere mapping. With a spherical map shape, when the viewpoint changes in a scene, the reflections sampled from the pre-defined images on the map will remain the same and appear wrong unless the entire environment map is recalculated "on-the-fly" (*Figures 5 and 6*). The developer's choices are bleak: create a sphere map one time for one orientation and view, or calculate millions of maps for every change in veiwpoint. Completely recalculating the sphere map for each frame or each time the viewpoint changes is prohibitively expensive, especially given the amount of effort needed to generate a single, satisfactory composite. With spherical environment mapping, even using today's state-of-the-art hardware, it is not possible to compute reflections interactively and in real-time.

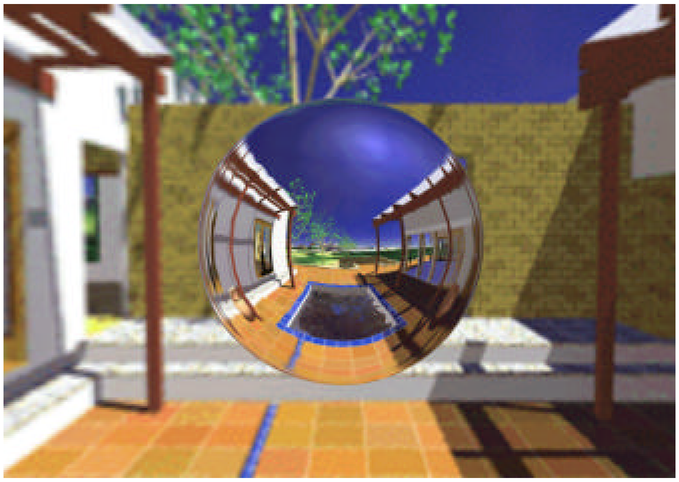

Figure 5: Spherical Environment Map
A correct reflection created with sphere environment mapping.

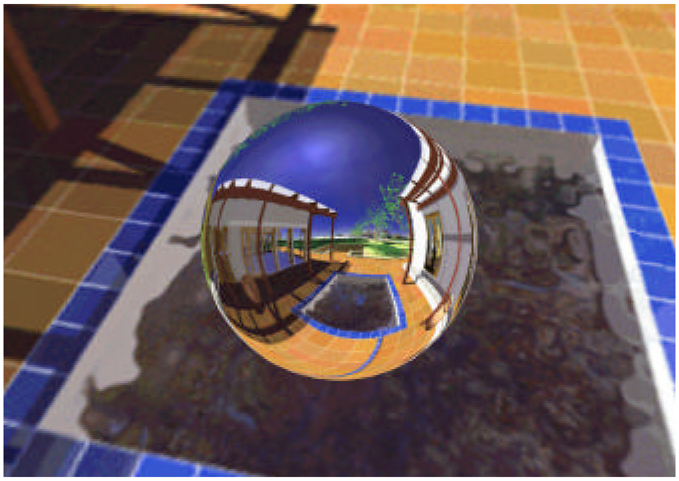

Figure 6: A Change in Viewpoint
The same sphere-mapped image with a change in viewpoint. The reflection is obviously incorrect. There is no way the viewer should see the pool and the reflection of the pool at the same time.

Costly in Human Effort and Computational Time

As noted above, with sphere mapping, designers need to spend a great amount of time creating a composite for the environment map for a reflection that will not be too objectionable when warped by the spherical map shape. The resources spent might be worth the effort if the resulting "reflection" was flexible enough to be used when the orientation or viewpoint changes. Yet, as a result of the combined computational expense and the risk of damaging image quality, environment mapping as it has been offered by some chip manufacturers, is largely ignored by developers.

Cube Environment Mapping:
A Simple Solution to a Complex Problem

Cube environment mapping in hardware is a breakthrough image quality feature of GeForce 256 that is fully supported by DirectX 7 and OpenGL and will allow developers to create accurate, real-time reflections. Accelerated in hardware, cube environment mapping will free up the creativity of developers to use reflections and specular lighting effects to create interesting, immersive environments. By changing the map shape to a six-sided cube, cube environment mapping offers a simple development path to the creation of stunning, reflective images.

With a cube map shape, reflections are captured by the six projected faces of the cube map that surround an object (*Figure 7*). This presents designers with a simple camera model that offers a path to the real-time generation of accurate reflections. A cube map shape allows linear mapping in all directions to six planar texture maps. The resulting reflection therefore does not undergo the warping nor damaging singularities associated with a sphere map, particularly at the edges of the "reflection."

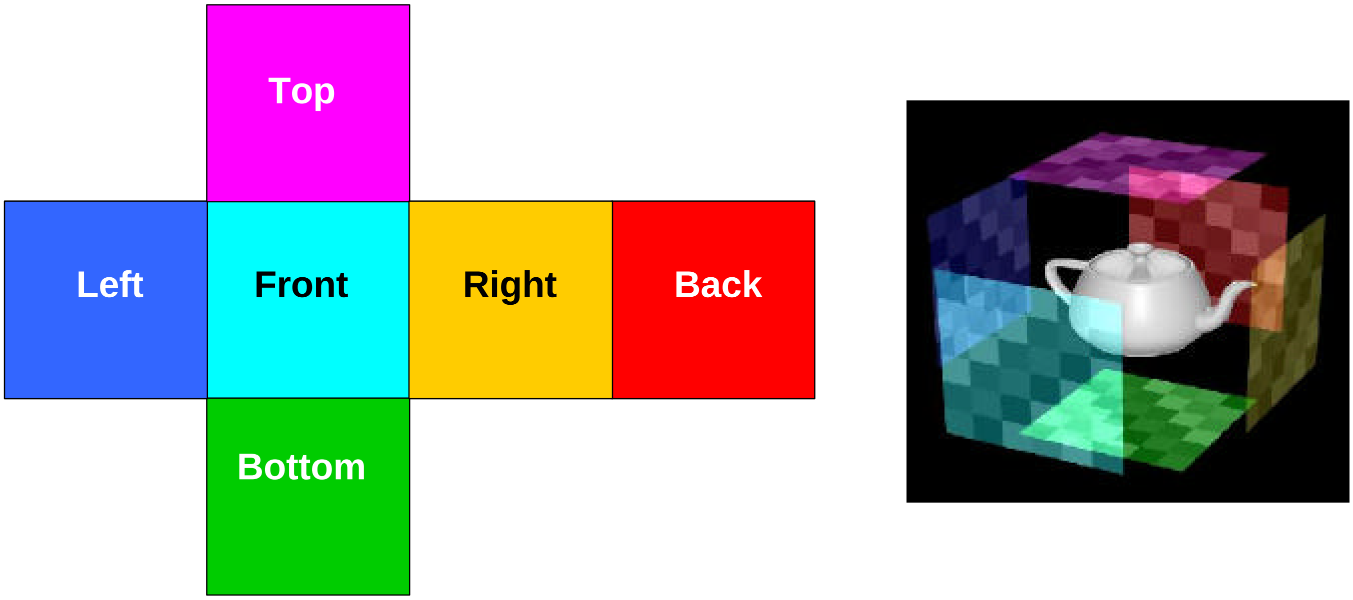

Figure 7: Cube Environment Map Shape *A cube environment map is created by stitching together six projective textures. The cube map shape allows for linear sampling in all directions for the secondary rays "reflected" off objects, eliminating the warping of previous approaches.*

Accurate Reflections

With a cube map shape, developers can take advantage of straightforward shots using standard camera modes to create reflection models without warping or singularities *(Figure 8).* Unlike sphere mapping which encodes the reflection function in image space that is correct for only one view, cube mapping encodes the reflection in all directions evenly. No pre-stretching or expensive reprojection steps are required as the GeForce 256 will accurately calculate the reflection vector and sample the correct planar environment map for every orientation. For the first time, creative uses of reflections to enhance game play can be employed without worrying about the risk of damaging the image quality of the scene.

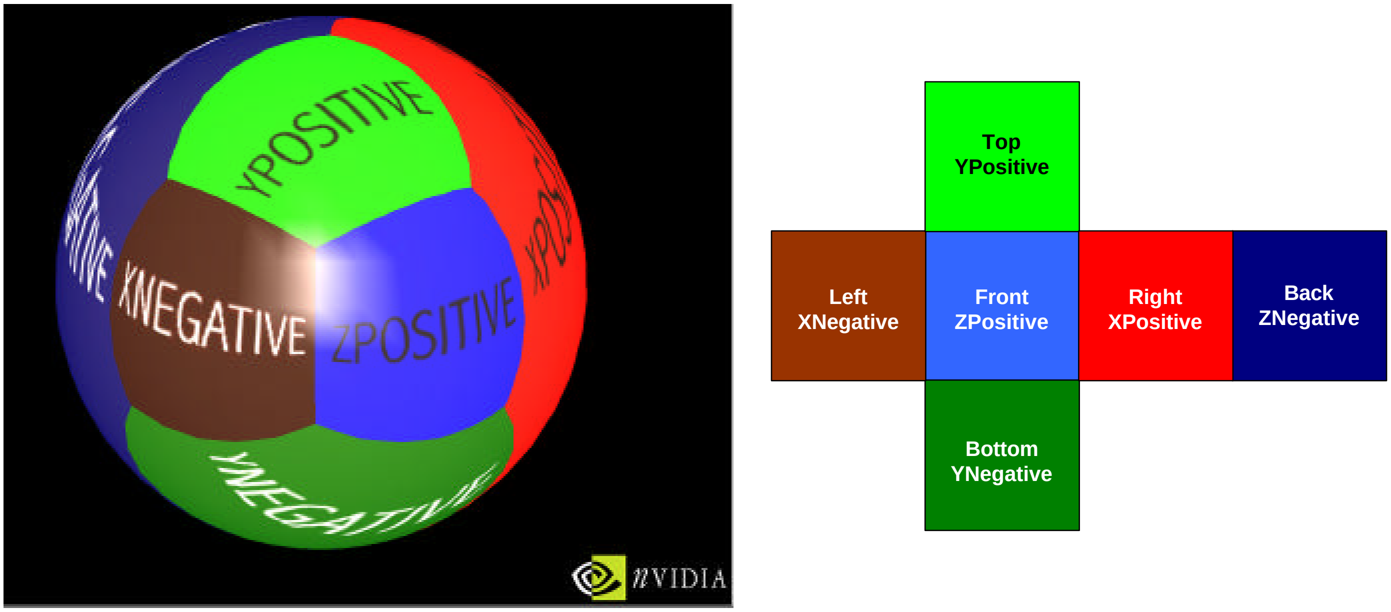

Figure 8: Cube Environment Mapped Sphere
The GeForce 256 rendered this sphere object on the left using accelerated cube environment mapping. The six planar texture maps that provide the lookup coordinates for accurately modelling reflected secondary rays are noted on the object. The "unfolded" cube map is on the right.

Viewpoint Independent and Real-time Drawing

The simplicity of cube environment mapping combined with the power of GeForce 256 acceleration, enables the rendering of real-time, viewpoint independent reflections. For any change in viewpoint, sphere maps typically require a recalculation for every frame. Cube environment maps, however, are viewpoint independent and in scenes with static surroundings do not have to be updated when the viewpoint moves. When the viewpoint changes, the reflection vectors are calculated by GeForce 256 and the correct planar environment map is sampled. In cases where designers wish to capture motion or changes in surrounding lights, the power of GeForce 256's hardware acceleration can be employed to update the map per frame as needed. Straightforward, standard camera shots enable real-time drawing of the reflection models.

Enabling A New Set Of Effects: Crisp Specular Lighting

Cube environment mapping is a new form of texture access, and is not just limited to simulating accurate reflections. The GeForce 256 with accelerated cube environment mapping can also look up textures given a normal or any application-specified vector, enabling a whole new set of graphical effects.

For example, cube environment mapping will be used to create realistic specular lighting highlights (*Figure 9*). Typically, simulating a glossy object like a billiard cue ball results in "seamed" specular lighting as vertex, per-polygon lighting is employed. By rendering specular lighting into the cube map, crisp *per-pixel* specular lighting effects can be created. Cube Mapping allows specular lighting lookups against the cube map on a per-pixel basis, enabling seamless specular highlight effects. The cube mapping technique gives developers lighting capabilities superior to per-pixel Phong shading with an arbitrary number of lights, including extended light sources. With a little creativity, a broad range of interesting, photorealistic effects can be created. For example, by blurring the cube map, objects that have rough surfaces such as brushed aluminum can be simulated. NVIDIA's GeForce 256 supports very large cube maps, up to 1K x 1K texels, for very high quality lighting effects.

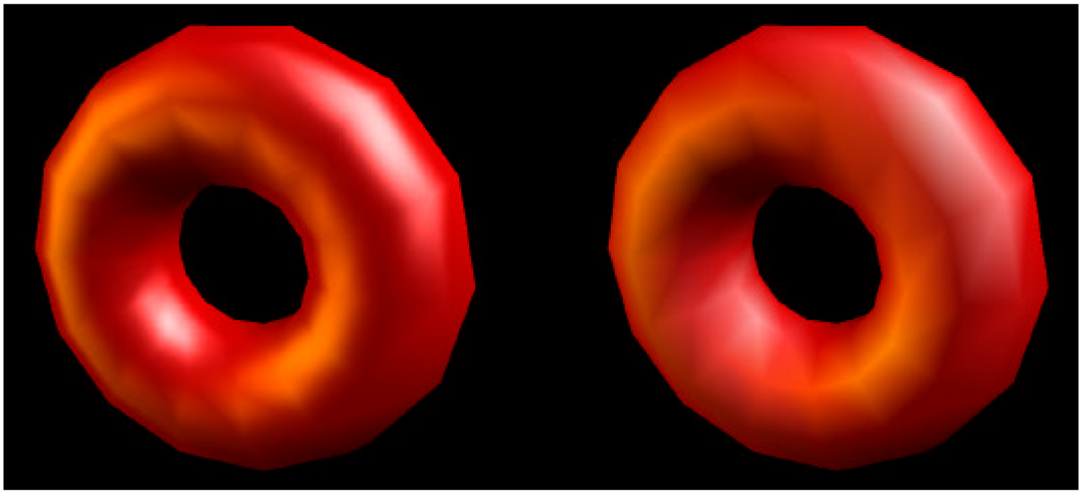

Figure 9: Per-Pixel Specular Lighting
On the left "donut", per-pixel lighting through cube mapping enables crisp, smooth specular highlights. The donut on the right uses vertex lighting. Notice the "banding" and edges that are clearly visible on the specular light on the upper right of the image.

Easy To Do

Cube environment mapping is an easily understood method. Unlike cumbersome sphere mapping where singularities and warping must be foreseen in advance, cube environment mapping is an easy-to-implement step toward photorealism that is cheap to generate in hardware. Essentially, all that is necessary is to point the camera, shoot, render and repeat six times for each side of the cube. The cube environment map is represented by a DirectDraw$^{\hat{a}}$ surface and can incorporate mip-maps for optimal speed. Developers are even free to choose the number of faces of the cube to sample against depending on the complexity of the image they wish to create, thus saving texture memory.

Accelerated in Hardware

What's more, the benefits of cube environment mapping are offered to the developer community "for free." Contrary to the software implementations of sphere mapping touted by competing chips, all secondary ray calculations generated by cube mapping technique are accelerated by the world's most powerful PC graphics processing unit, the GeForce 256. Stunningly accurate, real-time reflections and lighting effects without the performance penalties associated with the software approaches that rely on the CPU offer unmatched value to the end user.

An Industry Standard

Cube environment mapping is fully supported by DirectX 7 and OpenGL (please refer to NVIDIA's white paper, *"Microsoft® DirectX® 7: What's New for Graphics?",* to learn more about new DirectX 7 features). Soon to be a key differentiator in image quality and scene realism, game developers who are eager to fully unleash their creativity will utilize cube environment mapping for leading games for Fall 1999.

Developers Supporting Cube Environment Mapping:

-- GT Interactive
-- Epic MegaGames
-- LucasArts
-- Electronic Arts
-- Havas Interactive
-- Infogrames
-- Origin
-- Planet Moon
-- Cavedog Entertainment
-- Microsoft Games
-- Eidos Interactive
-- Blizzard Entertainment
-- Bullfrog
-- Activision
-- Ubi Soft
-- Monolith Productions
-- Acclaim
-- Rage Software
-- Bungie
-- Accolade
-- Codemasters
-- Valve
-- Lionhead Studios
-- Surreal Software
-- 989 Studios (Sony)
-- id Software
-- Raven Software
-- Ritual Entertainment
-- Nihilistic Software
-- 3DO
-- Sierra Studios
-- Gas Powered Games

Summary

NVIDIA's GeForce 256 delivers unmatched graphics performance and image quality, taking the visual experience on a desktop PC into stunning new territory. With features such as cube environment mapping in hardware, GeForce 256 will enable designers to unleash their creativity, opening up new possibilities for entertainment and game play. An industry standard that is easy to do, unrestricting on scene development, computationally cheap, and astonishing in image realism, cube environment mapping in hardware is a giant step toward the creation of immersive, realistic virtual environments on a desktop PC.