



nfiniteFX Engine Technical Brief

NVIDIA's 3D Texture Technology



*N*VIDIA®

The NVIDIA nfinite FX Engine

The development community is constantly looking for ways to improve the speed and flexibility of their games and applications. While visual effects such as lightning, volumetric fog, and explosions are all possible with a series of 2D textures, the implementations are not flexible and can create overhead due to texture swaps, API overhead, and a reduction in texture pipeline efficiency.

The NVIDIA® nfiniteFX™ Engine delivers the first true 3D texture technology aimed at consumers. NVIDIA's 3D texture technology allows for more flexibility, performance, and ease of implementation than traditional texturing techniques. The nfiniteFX engine can use a 3D texture from any angle or orientation (not just as a stack of 2D slices). As a result, NVIDIA's 3D texture technology gives developers the ability to create effects that cannot be implemented with a standard 2D texture approach.

What is a 3D texture?

A 3D texture contains information in three dimensions as opposed to two. The addition of the third dimension allows developers access to a depth component of texture information, as well as width and height. NVIDIA's 3D textures enables developer to think in new ways when texturing objects.

The image shown in Figure 1 is a fractal noise pattern in real-time (single pass) using four simultaneous 3D texture look-ups into a single 64x64x64 texture with tri-linear filtering

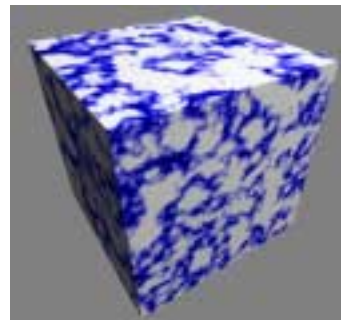
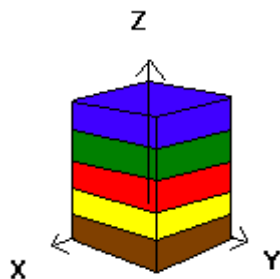


Figure 1. Fractal Noise Patter in Real Time

In the early days of 3D accelerators, the best technologies consisted of bilinear filtering of relatively low-resolution 2D texture maps. These 2D texture maps were mapped onto fairly large polygons, resulting in blurry textures and poor image quality. The technology continued to improve over time, allowing developers to use tri-linear filtering (bi-linear filtering of two different 2D textures) as well as Anisotropic Filtering (a technique used to improve the appearance of textures viewed from a very sharp angle).

Figure 2 shows a cut away of the marble block (fractal noise pattern). The nfiniteFX engine allows a slice of the 3D texture to be taken from any angle or orientation.

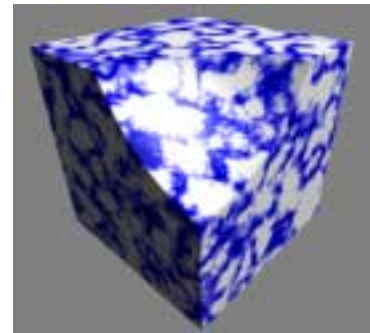
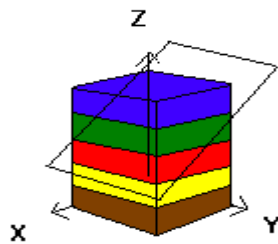


Figure 2. Fractal Noise Pattern with Slice of 3D Texture Taken Out

NVIDIA's 3D texture technology is the first to deliver the next step of quality through quad-linear filtering (tri-linear filtering of two 3D textures). Quad-linear filtering provides a feeling of depth and continuity so developers can fully simulate materials (wood, stone etc.) at higher levels of quality. The nfiniteFX engine empowers developers, allowing them to access and filter texture information from any angle or orientation.

3D Textures

The NVIDIA Advantage

Implementing 3D textures in a truly usable form is a unique but rewarding challenge. Here are some of the problems in implementing 3D textures along with NVIDIA's unique nfiniteFX engine solutions:

Problem: 3D Textures are Huge Files

A 256x256x256 texture would take 64MB uncompressed, while a 512x512x512 texture would consume 512MB uncompressed! This means that a 256x256x256 texture loaded completely into memory would consume the entire frame buffer of a 64MB board.

An effective compression scheme is essential to making large 3D textures usable. They need to be small enough so that can be moved quickly and efficiently between memory and the GPU.

Solution: NVIDIA's 3D Texture Technology

NVIDIA's 3D texture technology is the only implementation that supports compression in not just one or two, but three dimensions.

In addition, NVIDIA's compression technology supports multiple modes including multiple 4:1 compression modes and even an 8:1 compression mode! A 64MB texture can be reduced to 8MB. NVIDIA's 3D texture compression delivers a texture size that is manageable and usable.

Problem:
3D Textures Look Bad



Figure 3. 3D Procedural Texture without 3D MIP Mapping

Solution:
NVIDIA's 3D MIP Mapping



Figure 4. 3D Procedural Texture with 3D MIP Mapping

When mapping textures to polygons, it is often difficult to find a source texture that will closely match to the size of the resultant polygon without a lot of distortion. To solve this problem, many graphics applications contain multiple resolutions and sizes of any given texture. In determining what color a pixel should be, two different resolutions (MIP maps) of the source texture are chosen. Each of these textures are sampled at the appropriate locations and filtered. Those two results are then filtered together.

Images without MIP mapping alias and do not look as good as those that are MIP mapped, and 2D MIP mapping is not the same as MIP mapping in 3 dimensions (This is because the 3D level of detail will not be correct).

NVIDIA's Quad-linear filtering delivers higher image quality through true 3D MIP mapping. Textures are accurate (in 3D) at any angle or orientation.

Problem: 3D Textures Can be Slow

Solution: NVIDIA's 3D Texture Technology Improves Cache Efficiency

Texture locality is a big performance issue. If the texel you are looking for isn't in the texture cache, valuable time is wasted transferring in new chunks from memory. When 3D textures are minified (shrunk) and not MIP-mapped, adjacent screen pixels will tend to refer to texels that are not adjacent in the source texture. This will cause inefficiencies in the texture cache resulting in a much longer time to fill the triangle. Fortunately, NVIDIA's unique three-dimensional MIP Mapping reduces and filters the texture in a way that actually improves texture locality. In addition, the nfiniteFX engine's proprietary 3D compression takes advantage of specific attributes of 3D textures that allow more efficient cache usage.

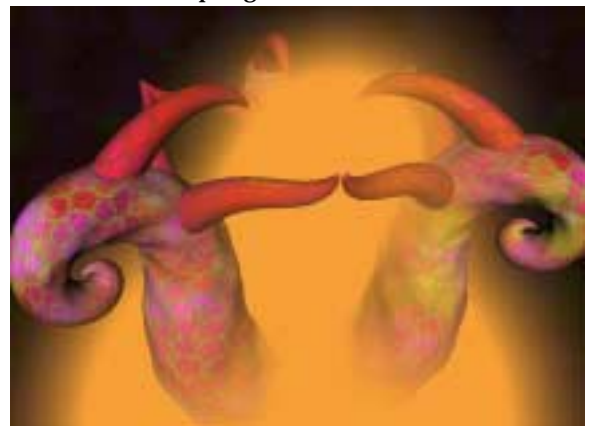
What Can nfiniteFX Engine 3D Textures Do?

The NVIDIA nfiniteFX Engine 3D Texture can produce the following effects:

- ❑ Volumetric Fog
- ❑ Imposters
- ❑ Function Lookup
- ❑ Procedural Textures and Noise

Volumetric Fog

Volumetric Fog is an effect that developers have been attempting to do for some time. By using the NVIDIA nfiniteFX 3D textures, developers can now create banks of fog that vary in density, not only in width and height, but in depth as well.



Imposters

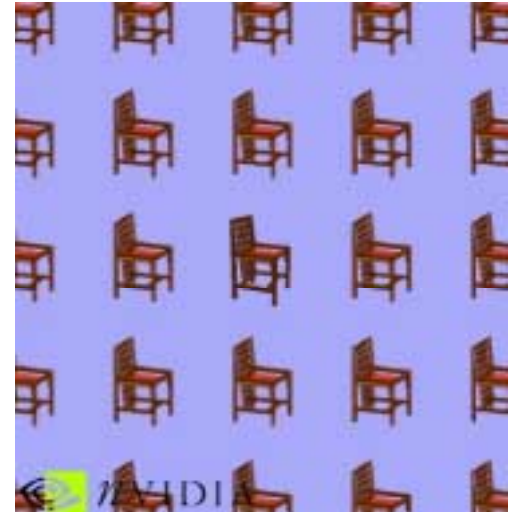
Processing models can consume a lot of CPU and GPU time. For each primitive in a model, every vertex must be transformed and lit. Pixels must be calculated, filtered, bumped and tested to see if it should even be rendered into the frame buffer.

This rendering process becomes a law of diminishing return for objects that are very far away from the camera. More and more calculations take place for objects that may only result in a few pixels on the screen. 3D textures provide an efficient solution to this problem. This solution is the use of Imposters.

Imposters are 2D primitives that remain orthogonal (directly facing) to the camera. A single 2D texture is then applied to the primitive. This gives the appearance that the application has transformed, lit, and calculated every vertex and pixel of a more complicated model, when in fact it has merely applied a single texture (which is much faster). The problem with this type of approach (only using a 2D texture) occurs if the object or light source is moving. In that instance, the single 2D texture will stand out and not look right.

It is in this situation where NVIDIA's 3D textures really shine. As a result of NVIDIA's 3D MIP Mapping, Imposters will look better (higher quality) as it grows and shrinks in the distance.

NVIDIA's 3D textures allow for the perception of correct look and orientation when the camera moves without excessive calculation.



As shown in this illustration, the Chair in the middle is real. The others are not.

Function Lookup

NVIDIA's 3D textures can store other types of information as well as texture data. 3D or 4D equations (with a variable held constant) can be looked up, as opposed to being calculated. This saves both time and effort.

One example is the Depth of Field effect (image on right). The 3D textures of the nfiniteFX engine can be used to contain information based on the distance from the camera, focal length, f-stop and the focus distance. As long as we hold one of these variables constant (f-stop for instance), we can easily implement this effect using nfiniteFX 3D textures.



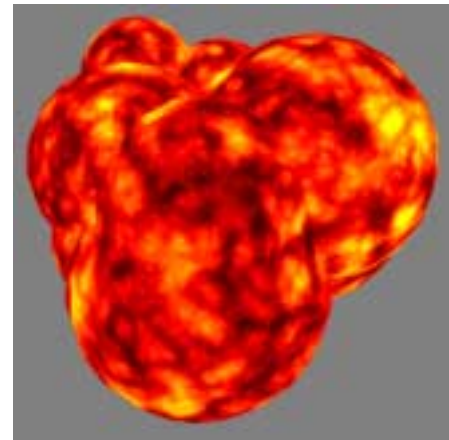
Another example of a function lookup using NVIDIA's 3D textures is to approximate a BRDF equation.

BRDF stands for bi-directional reflectance distribution function. BRDFs are used for a class of lighting where the reflected light off an object is not just a vector (or ray) that is equal to the angle of incidence (like a mirror). BRDF equations model a distribution around the angle of reflectance that is based on the material that the light is striking.

What makes this so interesting is that the material has imperfections that the naked eye can't see. Reflections off of the material will show these imperfections. The look of the object will vary based on the location of the viewer and the light source. A terrific example of this would be a piece of brushed metal, which would look different when viewed from different angles.

Procedural Textures and Noise

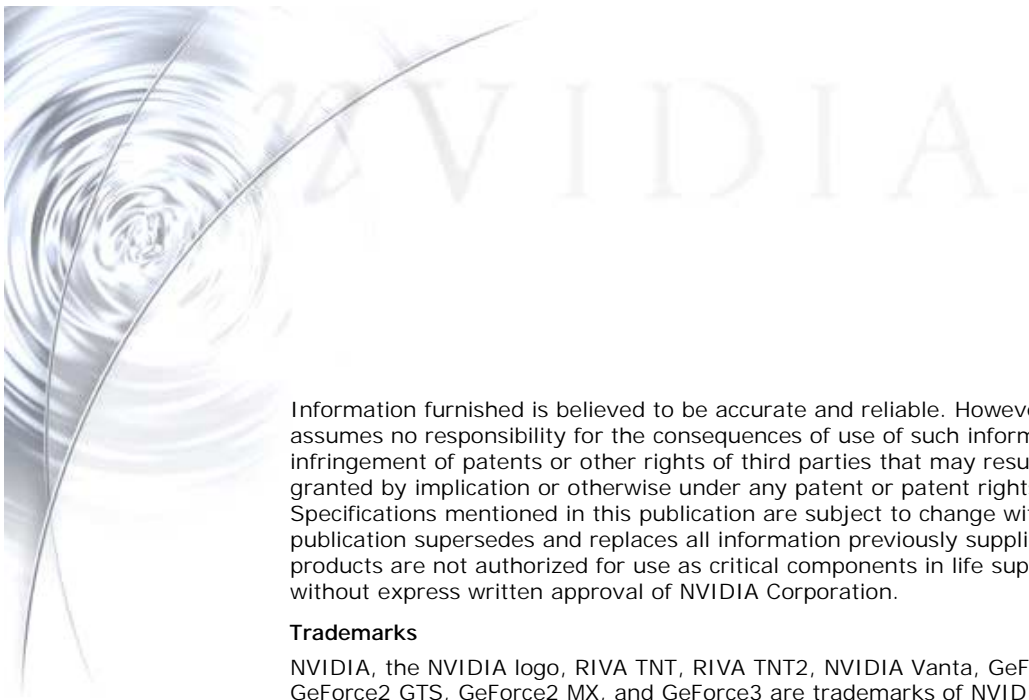
NVIDIA's 3D textures can also be used to create various procedural effects and noise. An example would be explosions (similar to the Perlin noise effects used in movies), lightning or plasma effects.



Conclusion

NVIDIA's 3D textures provide the first usable implementation of 3D textures for consumers.

With NVIDIA's unique 3D MIP Mapping technology, developers can implement quad-linear filtering to make their images higher quality. With NVIDIA's 3D texture compression, developers receive textures that are small enough to actually be useful. Lastly, the performance advantages of 3D MIP mapping, 3D compression and LightSpeed Memory Architecture deliver a fast and flexible solution that enables developers to solve many difficult graphics problems using 3D textures.



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, the NVIDIA logo, RIVA TNT, RIVA TNT2, NVIDIA Vanta, GeForce, GeForce2, and GeForce2 GTS, GeForce2 MX, and GeForce3 are trademarks of NVIDIA Corporation. Panel Link and TMDS are trademarks of Silicon Image, Inc.

Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation. RIVA, RIVA 128 and RIVA 128 ZX are trademarks of NVIDIA Corp. and STMicroelectronics. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

Copyright NVIDIA Corporation 2001.



NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050