FXT1: 3dfx Texture Compression

by Bubba "Masterfung" Wolford

Introduction

Every week it seems we are getting a new announcement about 3D technology. Recently, Nvidia and S3 both announced their chip specs.

While S3's Savage 2000 was taken with some skepticism at first because the specs looked so strong, others simply dismissed it as being another S3 offering that would never amount to anything, insisting that the rumored NV10 would, "blow it away." The next day, Nvidia announced their next generation chipset, the GeForce 256. Yesterday, over a conference call, 3dfx announced another new technology that will be implemented in their next generation chip.

3dfx Texture Compression: FXT1

My questions about FXT1 were very similar what you are all probably thinking right now: what is FXT1, what will it do for me and how is it different from what S3 has done with their own texture compression, S3TC?

FXT1 is a new technology that uses texture compression to compress textures using a four point encoding algorithm (as opposed to a single point algorithm used by S3) to condense textures to as much as 1/8th their size. The texture is coded into the game, flagged to be compressed by the hardware, and then compressed into memory using one of the 4 algorithms that decide the best way to encode the scene to minimize any loss in visual quality.

Next the texture is decompressed after identifying which algorithm was used to compress the texture, and is rendered on the scene all the while improving framerate and leaving more memory bandwidth open to be used as storage for more textures or by texture mipmaps. This encoding and decoding is done through hardware allowing the card and CPU to continue rendering frames as fast as possible.

The following chart demonstrates the three 3dfx texture compression algorithms and how they might be implemented. You can also see here the S3TC algorithm. FXT1 provides the most accurate image reproduction when measuring the Root Mean Square error of each encoding algorithm. Without question

First off, let me give you an example of what kinds of advantages texture compression offers. Suppose we have a texture that is 2048x2048 and 32-bit per texel. That texture will require a whopping 16 Mbytes of total memory space to store it locally in the texture memory of the video card! Keep in mind that this is only a single high-resolution texture!

3dfx's FXT1 could compress this texture to a size of 2048 Kbytes and thus free all the extra bandwidth for storing more textures! Since these textures are stored in the local memory of the video card, we have almost instantaneous framerate transfer and much higher fill-rates from the video card to your monitor!

Since 3dfx is using greater compression than S3's S3TC, we can look at this chart and see how 3dfx's FXT1 will be compressing textures:

Texture Size	8-bit	16-bit	24-bit	32-bit	4-bit FXT1™
64x64	6 KB	9 KB	13 KB	16 KB	2KB
128x128	18 KB	33 KB	49 KB	64 KB	8KB
256x256	66 KB	129 KB	193 KB	256 KB	32KB
512x512	258 KB	513 KB	769 KB	1024 KB	128KB
1024x1024	1,026 KB	2,049 KB	3,073 KB	4096 KB	512KB
2048x2048	4,098 KB	8,193 KB	12,289 KB	16384 KB	2048KB

Another advantage to 3dfx's texture compression is that it will be FREE to anyone. The code (open set source code) will be released to the public and is available to anyone. 3dfx is encouraging everyone to use the code and perhaps even tell them how they can improve upon the code.

There will be no license or royalty fees charges to use the technology. Plus, the new technology will be open to any OS, Windows, Mac, Linux and, hopefully, BeOS. Of course, 3dfx is making FXT1 available to all three major API's as it will be standard on Glide and open to D3D and OpenGL.

Go to Part II: How It Works

Copyright © 1997, 1998, 1999 COMBATSIM.COM, INC. All Rights Reserved.

Last Updated September 14th, 1999