

OpenGL Accelerated Effects and Graphics in the Boris FX Product Family

A White Paper by Boris Yamnitsky

August 2009

Graphics display hardware is widely recognized for visual effects creation. Despite many limitations and technical challenges confronting developers, the raw image processing power of modern GPU engines is unmatched by any multiprocessing, general-purpose computer system. The engineering crew at Boris FX was one of the first in the video industry to recognize this power and harness it to create the most stunning visual effects for post-production professionals.

2D Effects

Many traditional 2D effects were re-programmed to use the graphics engine:

BCC Glitter



BCC Lens Flare



BCC Lightning



BCC LED



BCC Damaged TV



Most of these effects use the CPU for relevant geometrical calculations and the graphics processor for scaling and compositing of individual layers. Most of them run in real-time in low-overhead environments.

OpenGL processing lends itself well to floating point image formats with no increase in render time. HD images are easily handled by employing tiling rendering techniques.

3D Effects in Boris Blue

The challenges in rendering professional-quality visual effects stem from close coupling of graphics hardware with the computer display. Features taken for granted in pro video systems, such as alpha channel processing and anti-aliasing, require special implementation on the graphics card. The first Boris products to use OpenGL – Boris RED, Boris Graffiti, and Boris FX – employed the graphics card for on-screen preview only, falling back on software renders for the final quality output. The first product to use an all-GPU render pipe was Boris Blue which made extruded, spline-based objects a reality.



Boris Blue was initially released as a standalone Windows application supporting NVIDIA Quadro FX graphics engines. Over time, Blue was made into a plug-in for Adobe and Avid software applications. At the time of this writing, the Macintosh version is still in the works. Boris Blue was the first Boris product to require a programmable GPU, relying exclusively on shaders for features such as lights, shadows, materials, bumps, depth of field, and motion blur. It sported a number of shader-based 2D effects such as high-quality Chroma Keying and Film Look. Because of the hardware acceleration, many Boris Blue compositions run in real-time. We used NVIDIA's CG to

run vertex and geometry shaders. Initially, we used the PBO (Pixel Buffer Object) extension for off-screen renders and later FBO (Frame Buffer Object).

Advanced features supported by Boris Blue include Anti-aliasing, Soft Shadows, Motion Blur, and Depth of Field rendering. These features were implemented with our proprietary multi-pass shader technique.

Native 3D Effects in Boris Continuum Complete 6

The development of Boris Blue created an opportunity for other Boris plug-in products to leverage the processing power of graphics cards. The latest release of Boris Continuum Complete (BCC Version 6) became the first native plug-in set to use advanced GPU shaders for 3D creation within desktop compositing and editing applications. The BCC release required Mac OS support as well as support for a much wider array of graphics cards and drivers. As a plug-in, the program is severely restricted in resources such as texture memory; it may be called from multiple threads and it has no control over the graphics card or driver version.

Special Challenges

Resource management under Dual Screen Mode in Quadro FX cards has been particularly difficult. Off-screen render contexts appear to require double-sized memory buffers. Anti-aliasing implementation requires a rendering context with a huge amount of memory. Also of considerable difficulty is off-screen rendering to an anti-aliased (AA) memory buffer and displaying the frame in non-anti-aliased mode on screen.

Macintosh Products

Our Mac OpenGL technology predates CG on Mac. For that reason, and to better control quality, we compile CG code to ARB shader assembly on Windows and install them on Mac. Moreover, different shaders are compiled with different versions of the CG compiler. Encrypted ARB shaders are installed by our installer. We do not employ CGFX; instead, we use a proprietary system of parameter definitions.