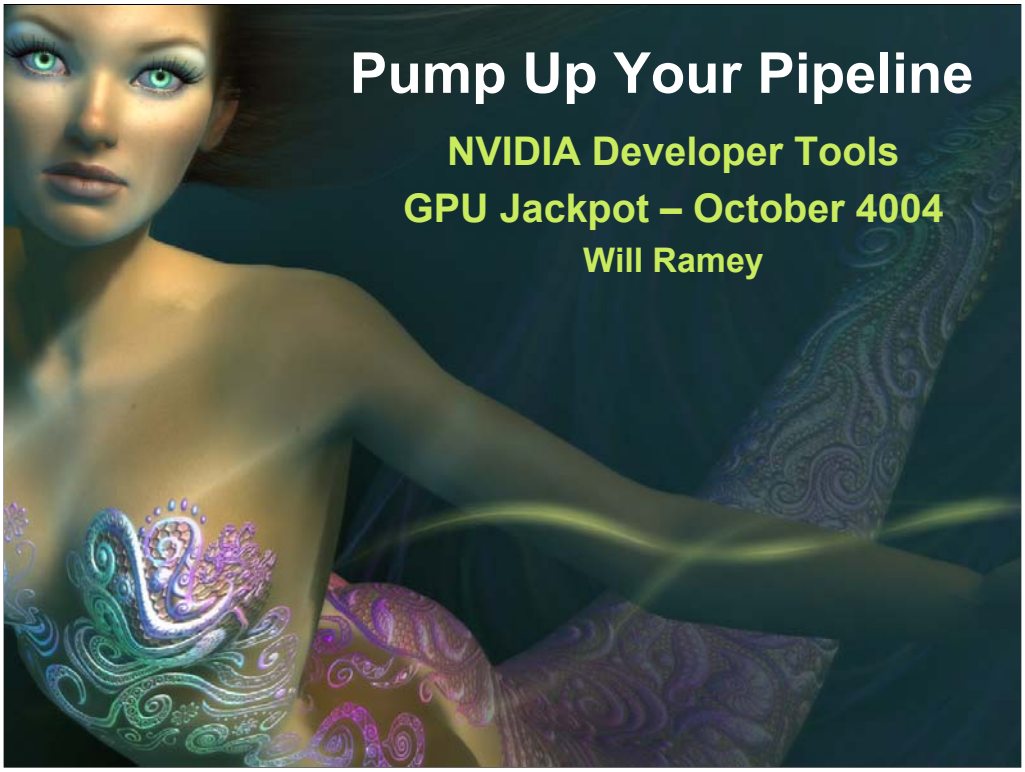

Pump Up Your Pipeline
NVIDIA Developer Tools
GPU Jackpot – October 4004
Will Ramey

Why Do We Do This?

- Investing in Developers Worldwide
  - Powerful tools for building games
    - Software Development
    - Content Creation
    - Performance Analysis
  - Practical SDK with technical documentation
  - Web Site and Newsletter developer.nvidia.com
- Registered Developer Program
  - Pre-Release Drivers
  - Early Access to Developer Tools
  - Online Support Forums & Bug Submission

Sign up now at developer.nvidia.com

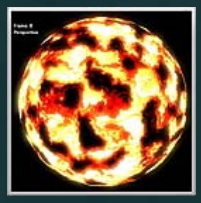
NVIDIA SDK

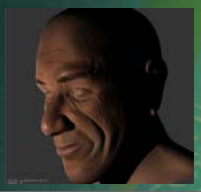
The Source for GPU Programming

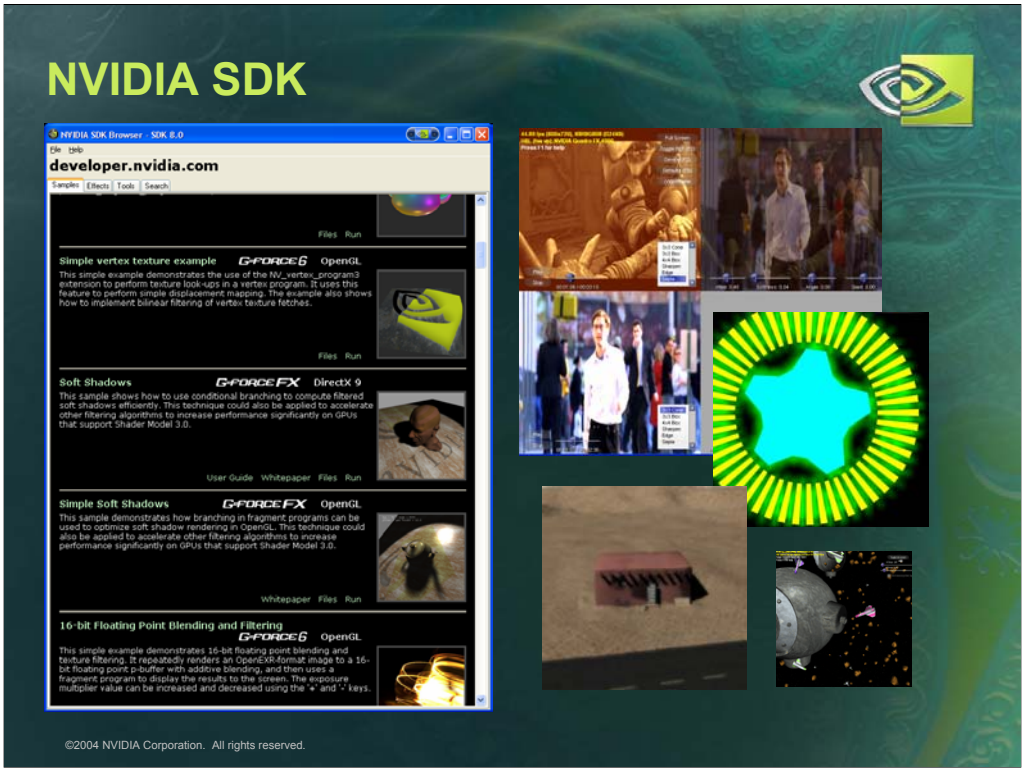
Hundreds of code samples and effects that help you take advantage of the latest in graphics technology.

- **Tons of updated and all-new DirectX and OpenGL code samples with full source code and helpful whitepapers:**
  - **Geometry Instancing, Rainbow Fogbow, Blood Shader, Perspective Shadow Maps, Texture Atlas Utility, ...**

- **Hundreds of effects, complete with custom geometry, animation and more:**
  - **Skin, Plastics, Flame/Fire, Glow, Gooch, Image Filters, HLSL Debugging Techniques, Texture BRDFs, Texture Displacements, HDR Tonemapping, and even a simple Ray Tracer!**





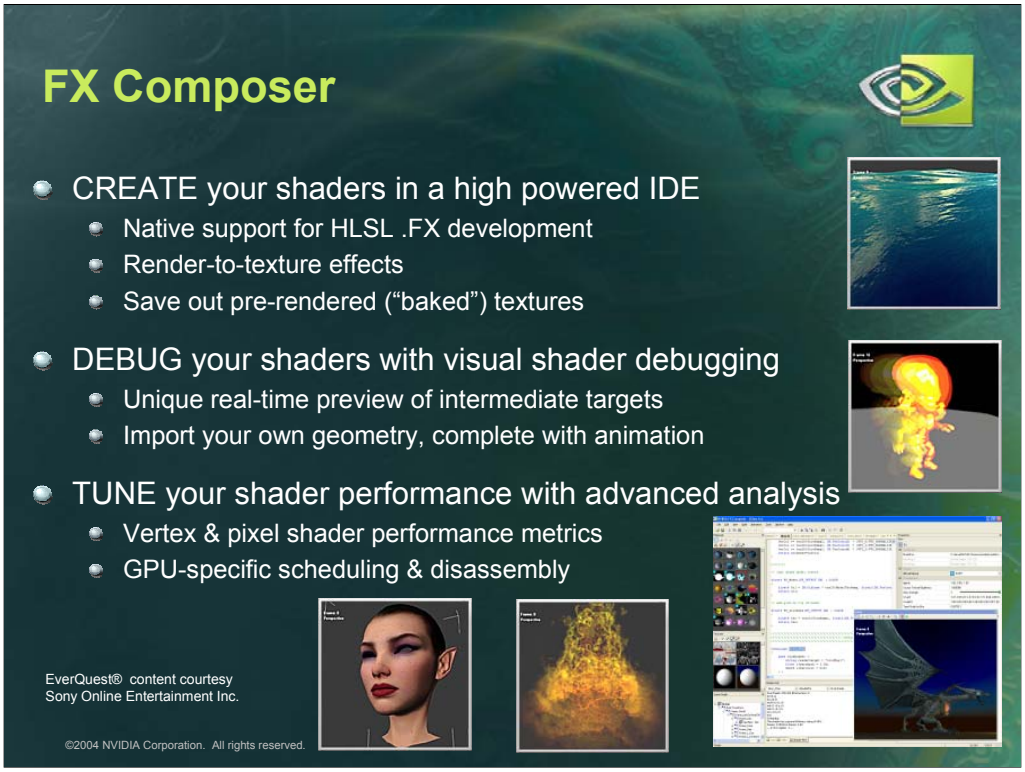


SDK Browser

Search, sort and find exactly what you need

Practical Code Samples & Compelling Effects

- Video Filtering
- Antialiasing with Post-processing
- Deferred Shading
- Geometry Instancing
- 250+ compelling effects

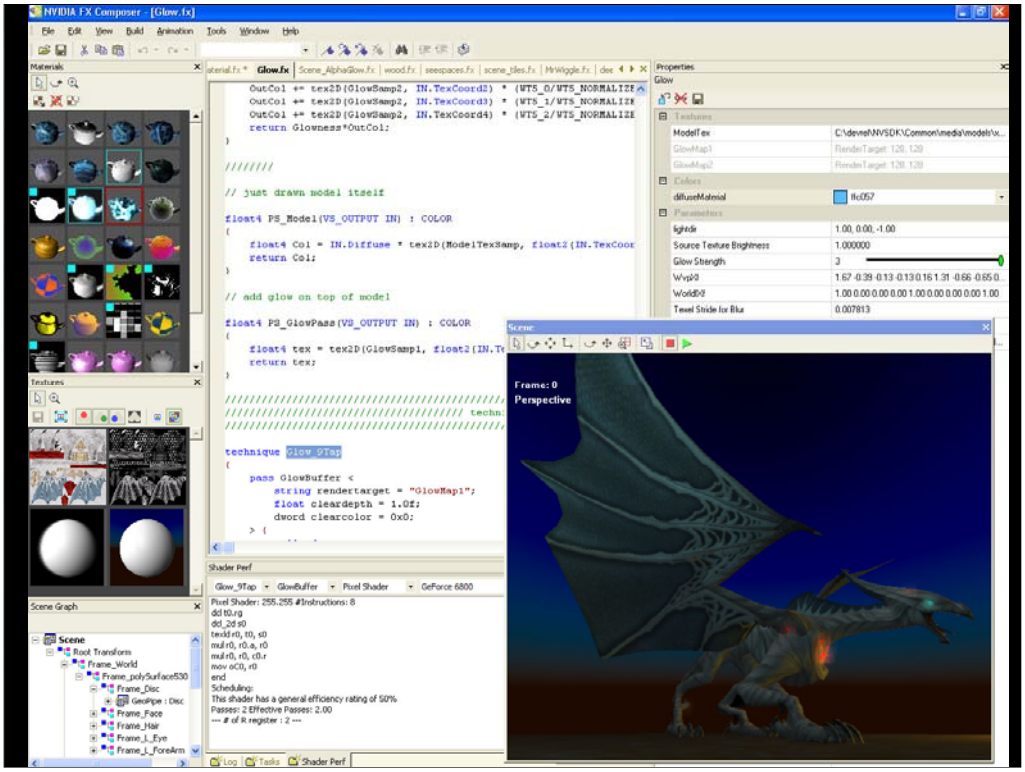


Be sure to check out the complete User Guide and helpful tutorials!

Also the full presentation dedicated to FX Composer.

EverQuest® content courtesy Sony Online Entertainment Inc.

EverQuest is a registered trademark of Sony Computer Entertainment America Inc.



CREATE your shaders in a high powered development environment

Sophisticated text editing with intellisense (auto-complete) and syntax highlighting

Work directly with HLSL .FX files, creating multiple techniques and passes

Use the .FX files you create with FX Composer directly in your application

Convenient, artist-friendly graphical editing of shader properties

Supports Microsoft DirectX standard HLSL semantics and annotations

Support for all DirectX 9.0 shader profiles

Develop your shaders on your own models with lighting and animation

DEBUG your shaders with visual shader debugging features

Interactive compiler helps you find and fix problems

Visible preview of source and intermediate textures targets

Interactive jump-to-error feature helps you fix problems quickly

TUNE your shader performance with advanced analysis and optimization

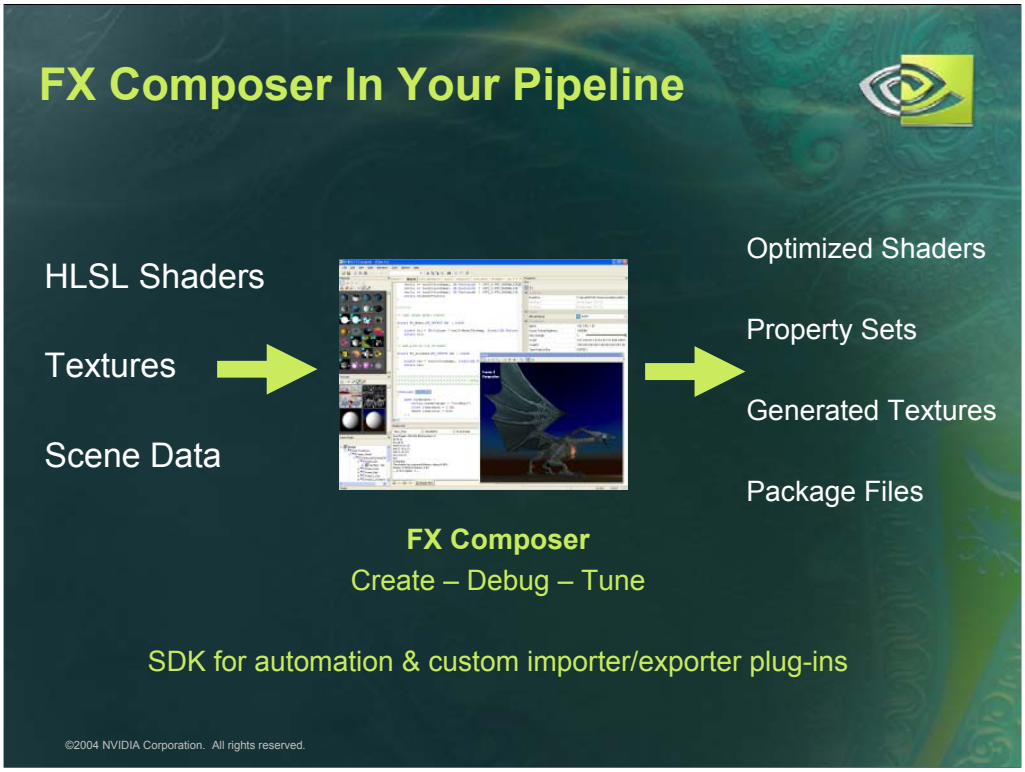
Enables performance tuning workflow for vertex and pixel shaders

Simulates performance for the entire family of GeForce FX GPUs

Capture of pre-calculated functions to texture look-up table

Provides empirical performance metrics such as GPU cycle count, register usage, utilization rating, and FPS.

Optimization hints notify you of performance bottlenecks



Goes-in-to

Supports all DirectX texture formats.

Import scene data from .obj, .x, .ply, or .x files.

Comes-out-of

Use the Shader Perf panel to optimize your shaders.

Save properties sets as XML files (re-apply in engine).

Save textures generated by HLSL functions (e.g. noise).

Package shaders, textures & full scene in one file for review/collaboration.

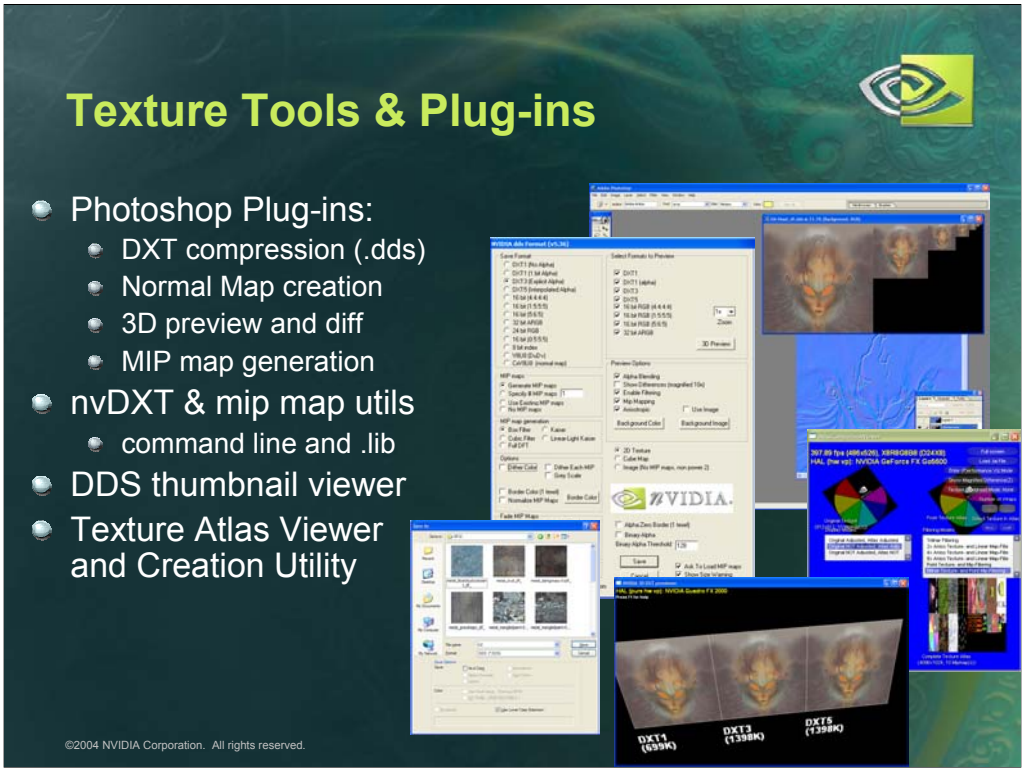
SDK for automation & custom importers/exporters

Sample code for custom importers & exporters

Save properties bundles in your own format

Automate just about anything using .NET scripting

(e.g. screenshots for all shaders in a directory)



Photoshop Plug-ins

Allows saving in various formats, incl. DXTC (.dds files)
Generate mip-maps in variety of ways

(box filter to Kaiser gamma-space filter)

Preview w/ 3D rendering options

(e.g. anisotropy, format comparisons etc.)

Create normal maps from height maps

nvDXT & mip map utilities

Command line interface for pipeline automation

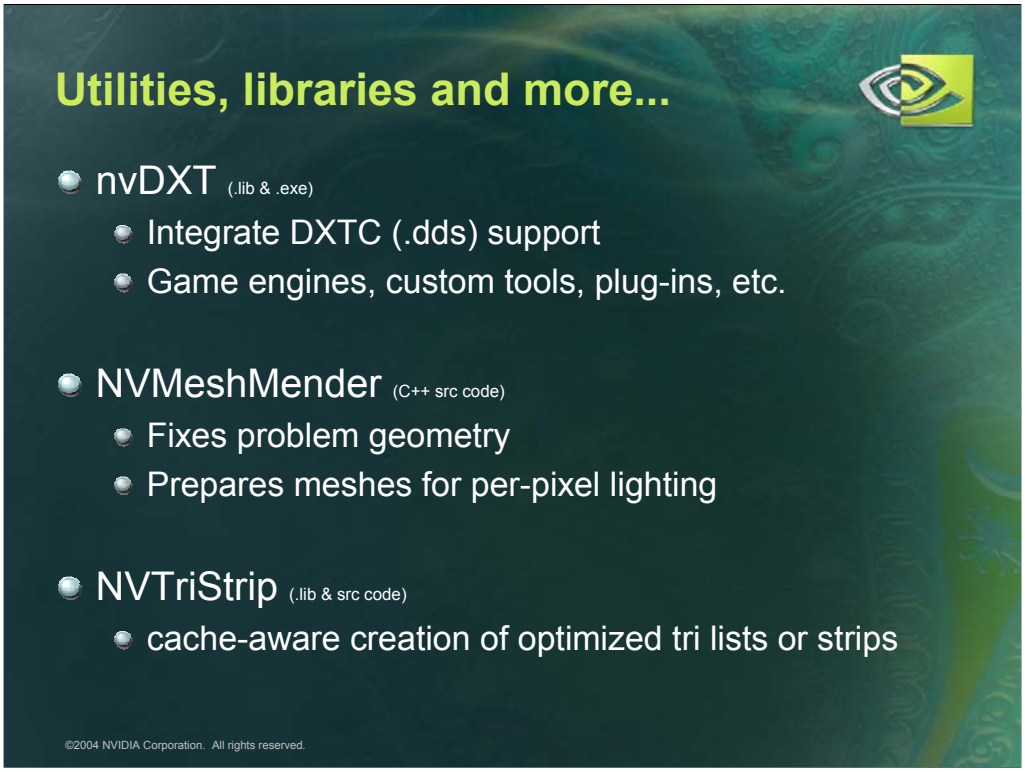
Available .lib for tools integration

Detatach & Stitch utils to manipulate mip maps

DDS Thumbnail Viewer

Provides convenient preview of DXT compressed files.

See Texture Atlas whitepaper for details...

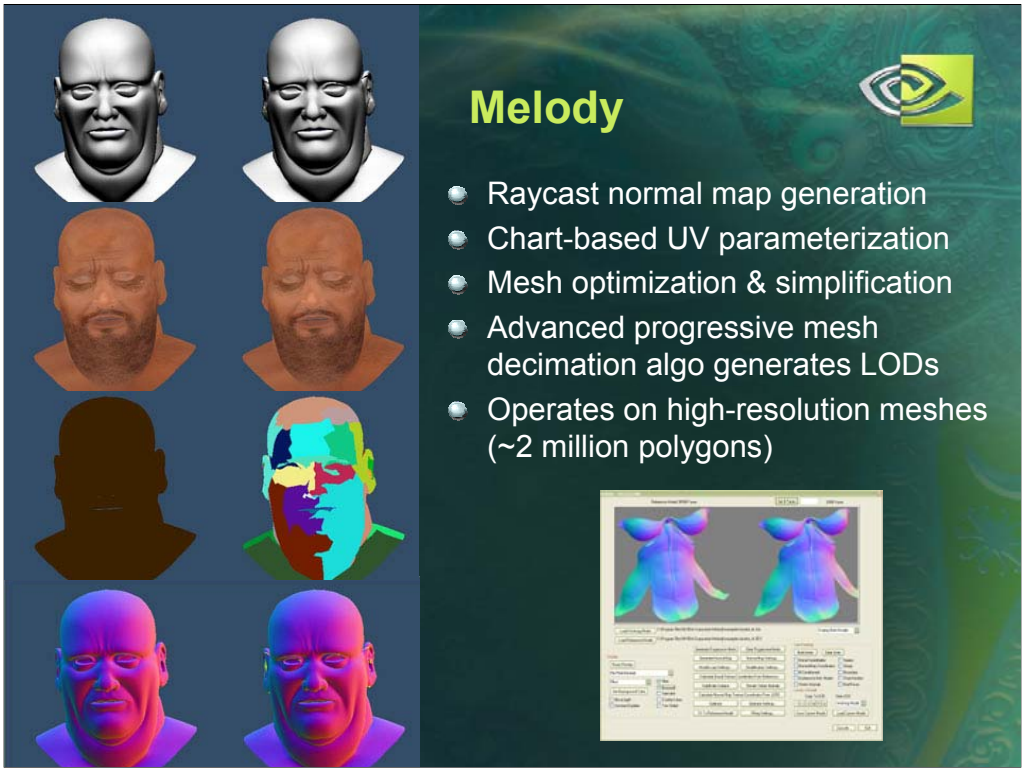


NVMeshMender:

Creates tangent basis for per-pixel lighting
Creates smoothed normals
Creates u, v coordinates
Handles mirrored (u,v) and cylindrical wrapping

NVTriStrip (library):

Library to stripify arbitrary geometry meshes
Flexibly optimizes for postTnL vertex caches
Option to stitch strips using degenerate triangles
Option to remap indices to improve vertex buffer locality
Outputs lists or strips



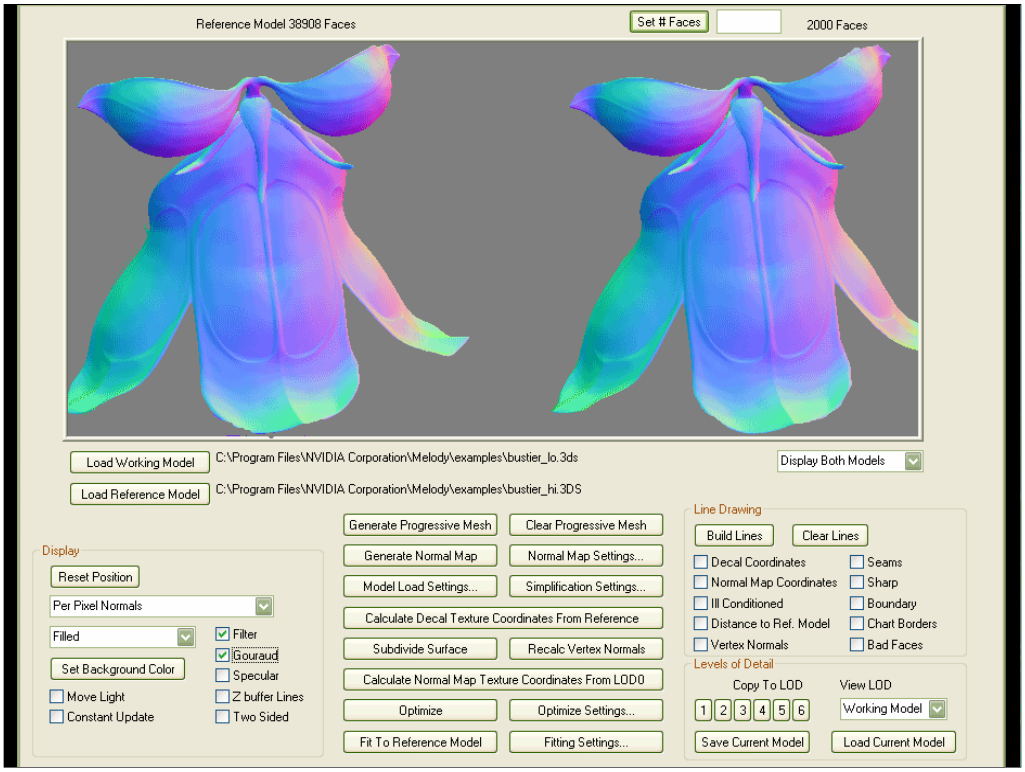
Head model (right side of slide)

Left: 18k poly reference model

Right: 4k poly working model (auto simplified using progressive mesh deformation algorithm)

Notice the different view modes on the right. From top to bottom:

1. Filled
2. Textured
3. Chart View
4. Object Space Normals



Dawn's Bustier

Left 38K polygons

Right 2K polygons with Melody normal map

Can you tell the difference?

Visualization Options Include:

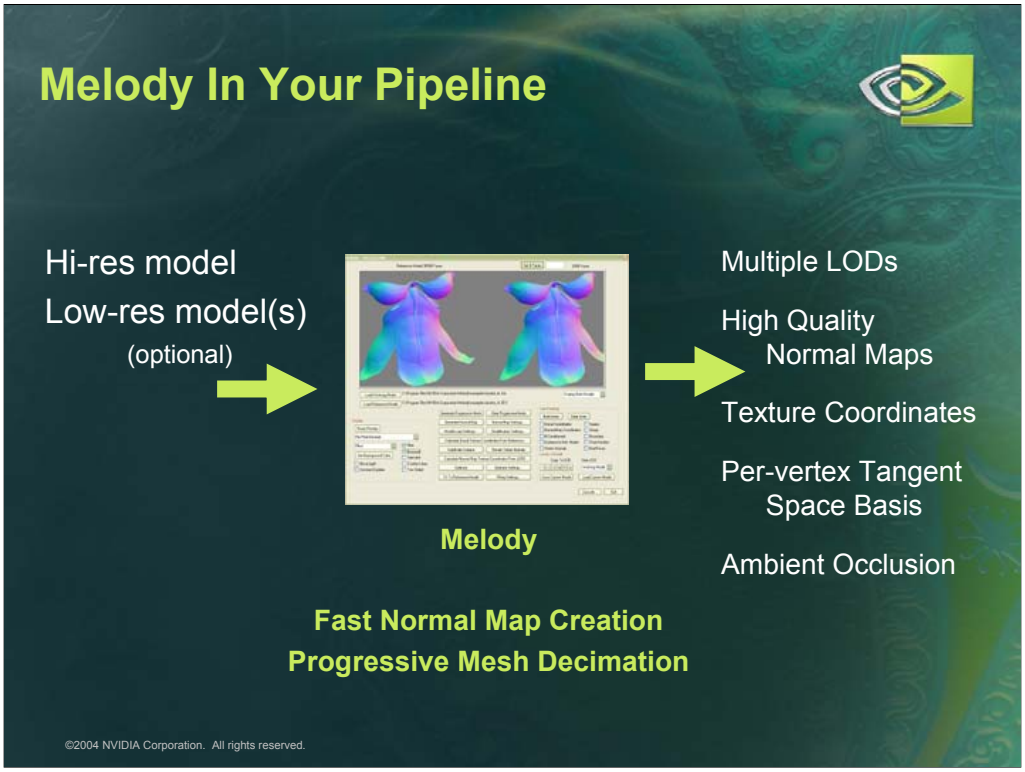
•Vertex or per-pixel normals (textured or not)

•Charts & attribute groups

•Tangent to object space

•Tangent or object space light mapping

•Combined with Filled, Wireframe or Outlined mode



Often takes longer to load the model than generate the normal map!

Goes-in-to

Import models from .obj, .3ds or .ply files.

Import a low-res model or generate LODs using Melody

Comes-out-of

Multiple LODs ()

Normal map file for each low-res model (tons of options)

Object space and tangent space normal maps

FP support (8 or 32 bits / channel)

Displacement maps

Re-sampled color map (normal & color)

Greyscale Height map

Generates ambient occlusion term

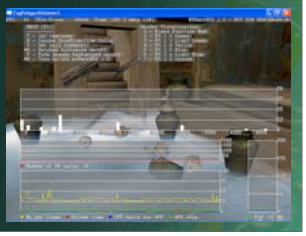
Visual Preview

Simple object manipulation

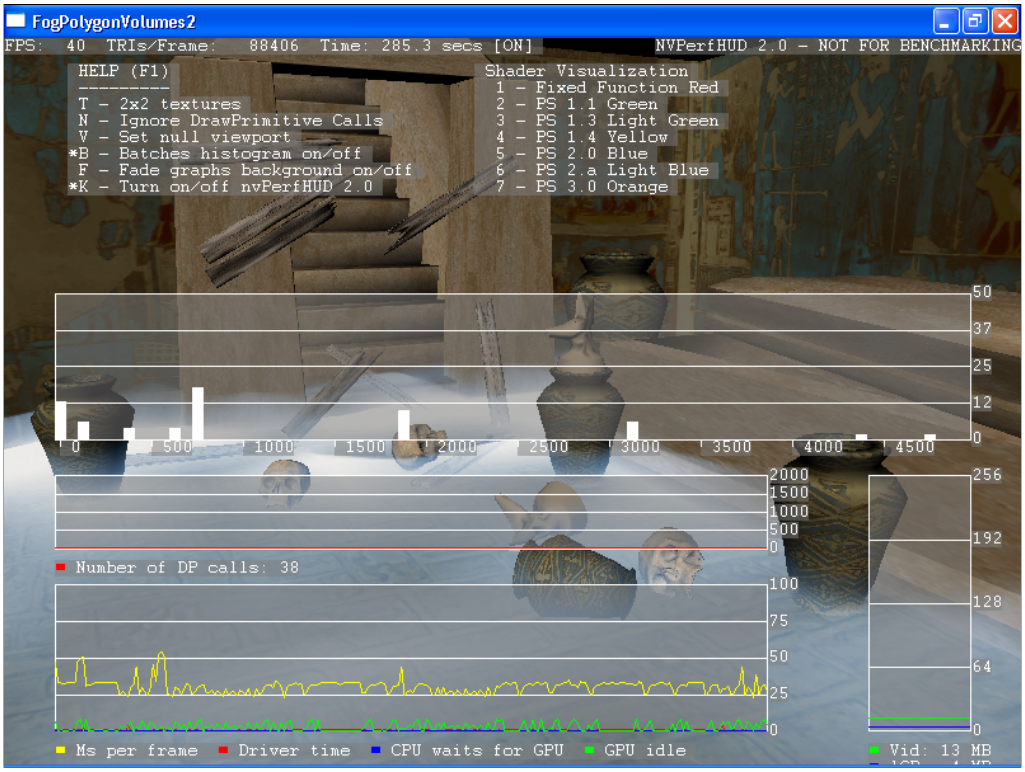
Powerful visualization modes

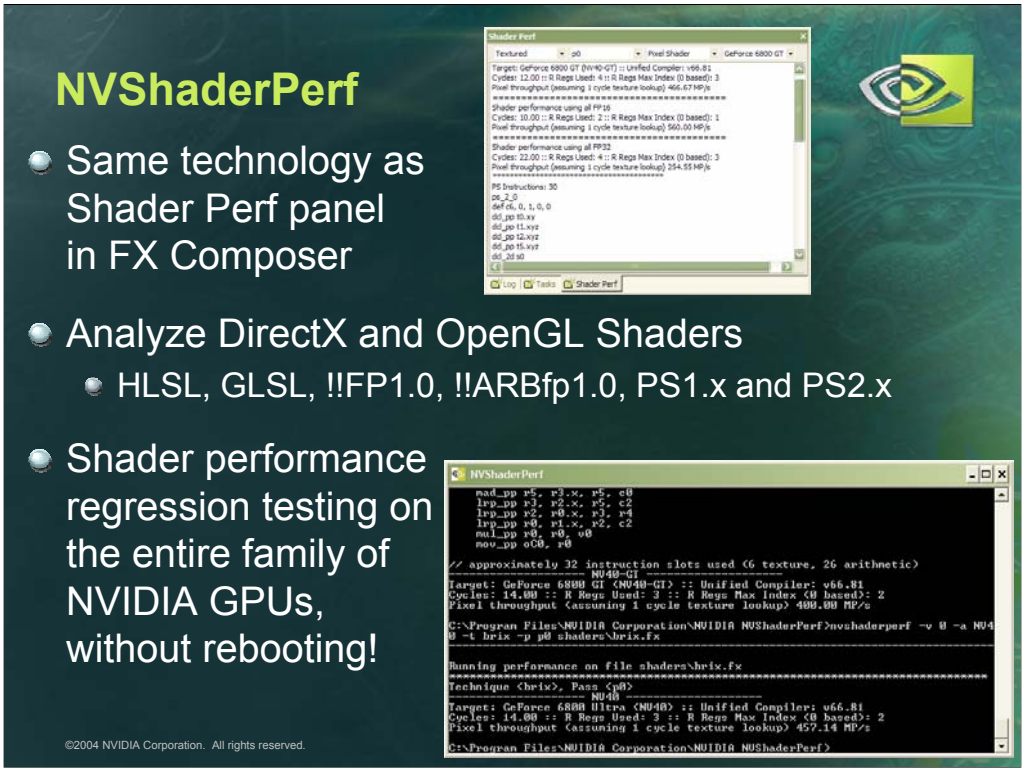
NVPerfHUD

- Graph overlay of various vital statistics
  - Shown on top of your running application
- Metrics reported include:
  - # DIP/DP calls per frame and batch size histogram
  - GPU_Idle – Graphics HW not processing anything
  - Driver_Waiting – Driver waiting for HW to finish
  - Time_in_Driver – Total time CPU is executing driver code
  - Frame_Time – ms between frame start & next frame start
  - AGP/Video memory usage
- Supports any Direct3D9 application



Be sure to check out the full User Guide!


FogPolygonVolumes2
FPS: 40 TRIs/Frame: 88406 Time: 285.3 secs [ON]
NVPerfHUD 2.0 - NOT FOR BENCHMARKING
HELP (F1)
T - 2x2 textures
N - Ignore DrawPrimitive Calls
V - Set null viewport
*B - Batches histogram on/off
F - Fade graphs background on/off
*K - Turn on/off nvPerfHUD 2.0
Shader Visualization
1 - Fixed Function Red
2 - PS 1.1 Green
3 - PS 1.3 Light Green
4 - PS 1.4 Yellow
5 - PS 2.0 Blue
6 - PS 2.a Light Blue
7 - PS 3.0 Orange
50
37
25
12
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
2000
1500
1000
500
0
256
192
128
64
0
Number of DP calls: 38
100
75
50
25
0
Ms per frame
Driver time
CPU waits for GPU
GPU idle
Vid: 13 MB



Different version for each version of NVIDIA drivers.

Identify shader performance issues:

Run once across all your shaders to create a baseline report.

Run the same version again later to identify which shaders have improved / degraded performance.

Identify driver performance issues:

Run once across all your shaders to create a baseline report.

Then run a different version on the same shaders.

improvements and/or regressions.

Let us know if you find any driver performance issues!!

Questions / Feature Requests?

All of this and more, available now at no cost to you!

developer.nvidia.com

The Source for GPU Programming

Please send questions, feature requests & comments about our SDK and developer tools to:

sdkfeedback@nvidia.com