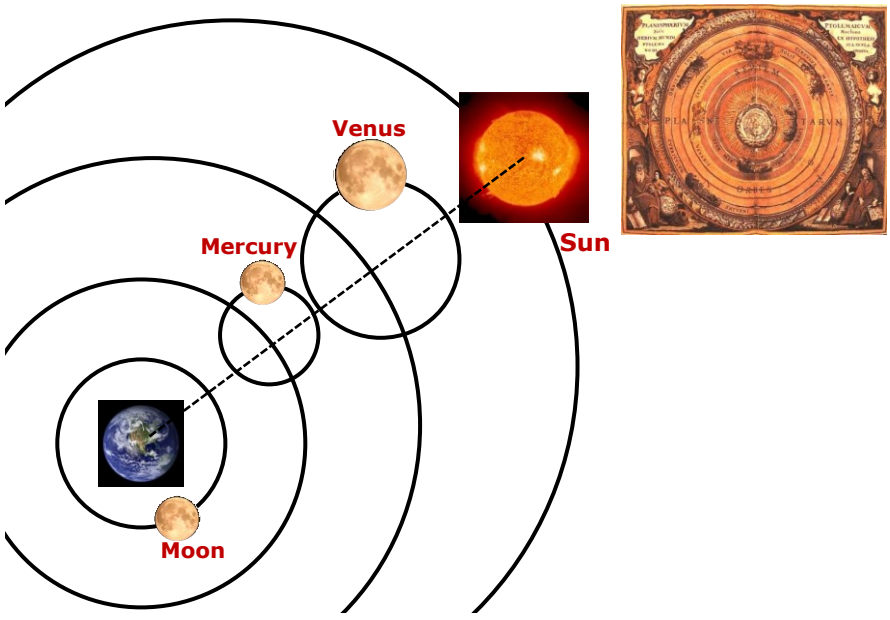
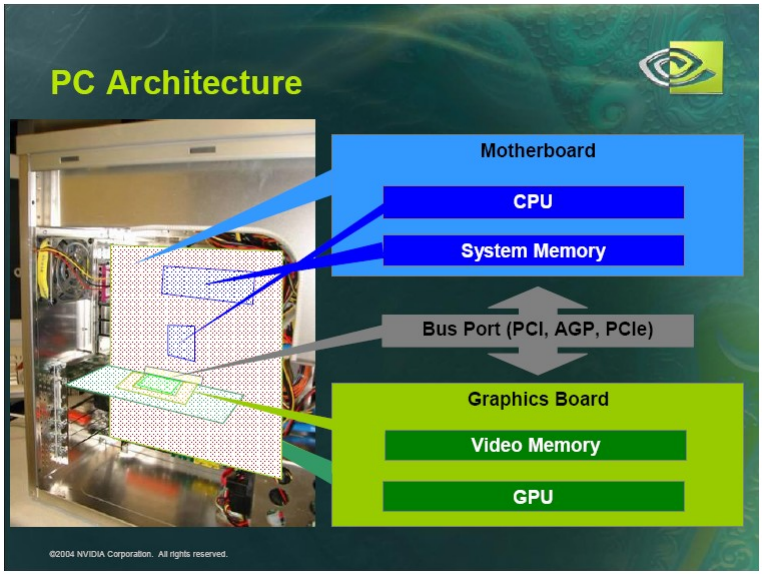


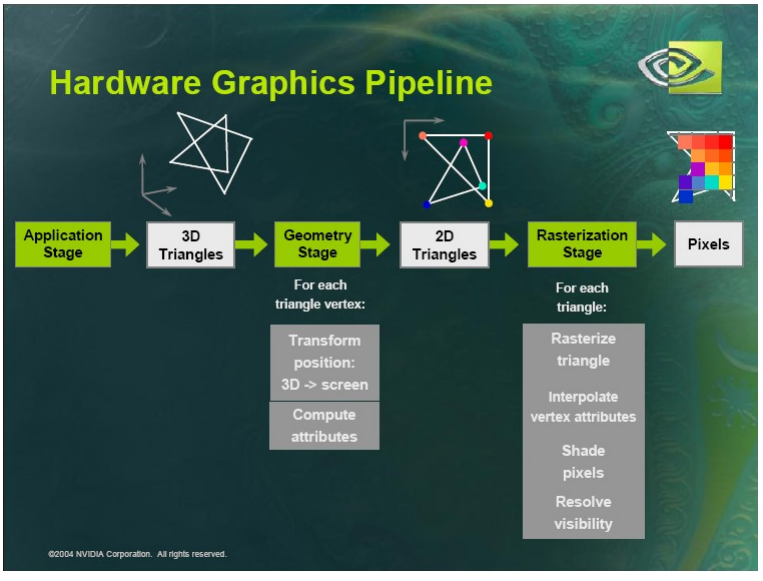
Graphics Processing

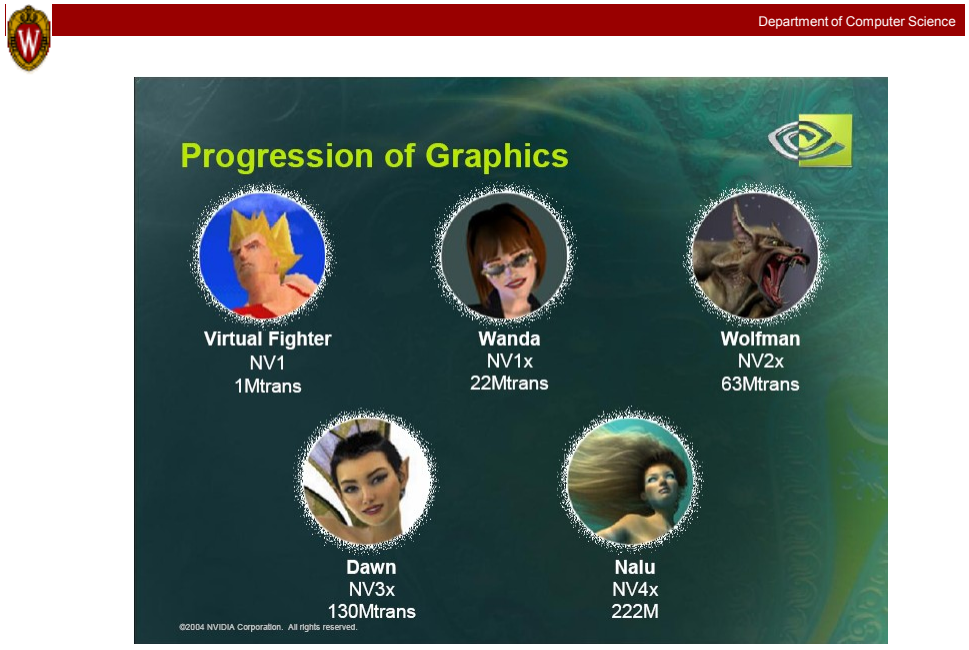
Karu Sankaralingam
Vertical Research Group
http://www.cs.wisc.edu/vertical

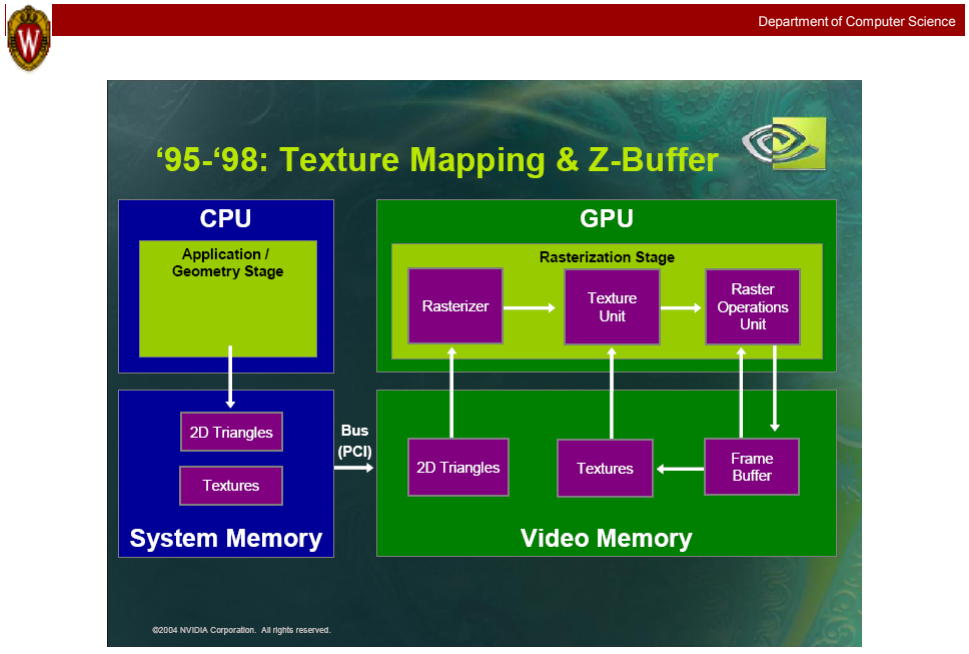
Collaborators: Venkatraman Govindaraju, Peter Djeu, Mary Vernon, William R. Mark

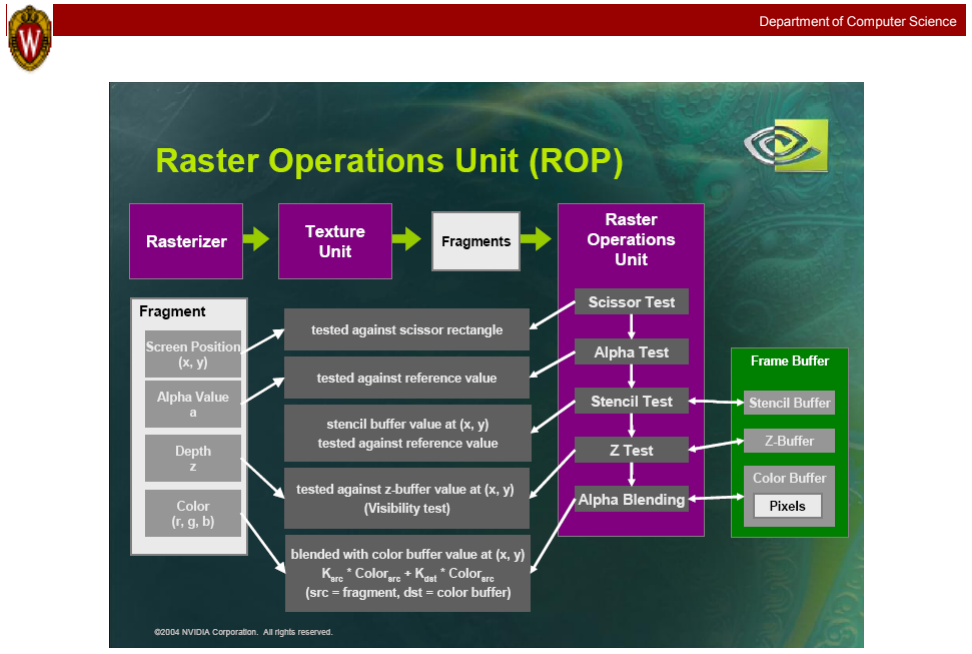
Evolution of GPUs slides courtesy of David Luebke and Charles Seitz

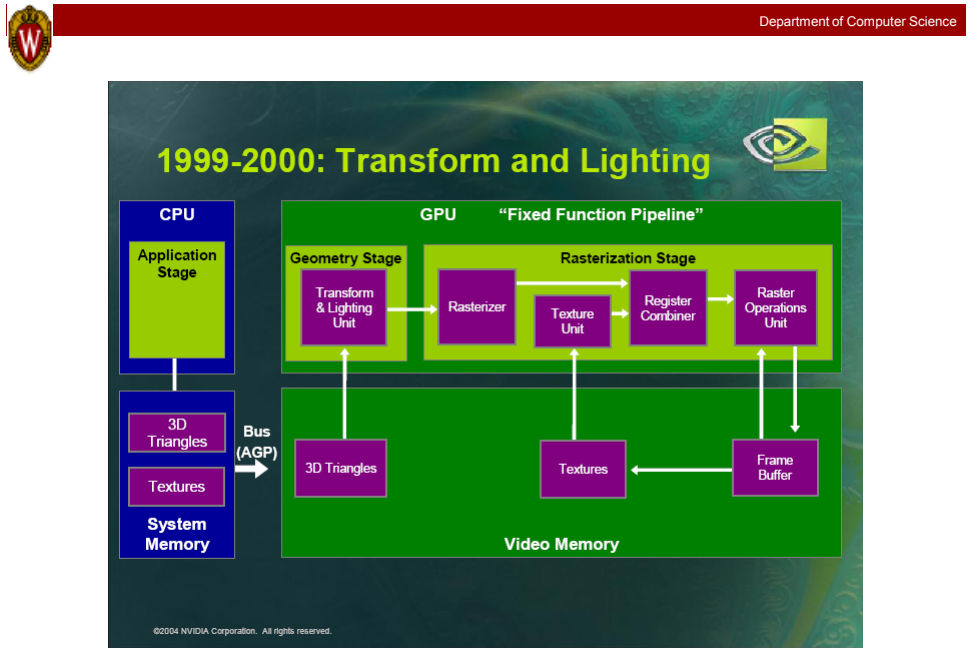

PC Architecture
Motherboard
CPU
System Memory
Bus Port (PCI, AGP, PCIe)
Graphics Board
Video Memory
GPU
©2004 NVIDIA Corporation. All rights reserved.


Hardware Graphics Pipeline
Application Stage
3D Triangles
Geometry Stage
2D Triangles
Rasterization Stage
Pixels
For each triangle vertex:
Transform position: 3D -> screen
Compute attributes
For each triangle:
Rasterize triangle
Interpolate vertex attributes
Shade pixels
Resolve visibility
©2004 NVIDIA Corporation. All rights reserved.

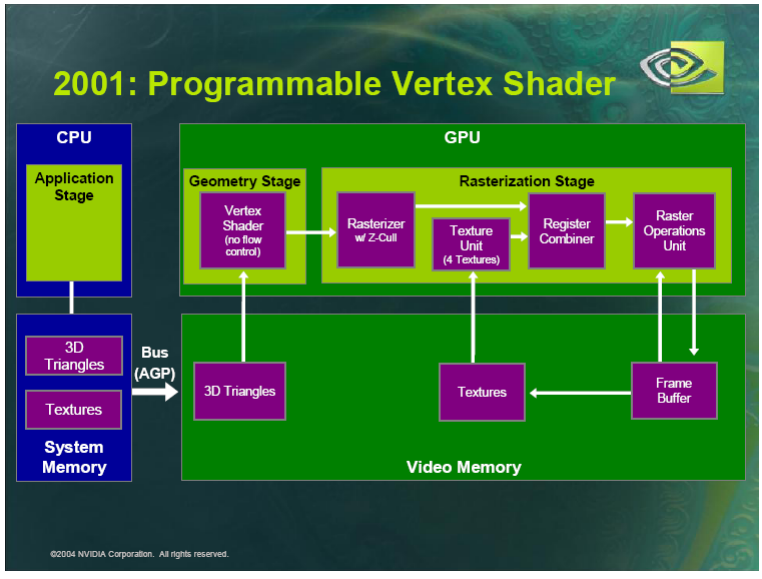

Department of Computer Science
Progression of Graphics
Virtual Fighter
NV1
1Mtrans
Wanda
NV1x
22Mtrans
Wolfman
NV2x
63Mtrans
Dawn
NV3x
130Mtrans
Nalu
NV4x
222M
©2004 NVIDIA Corporation. All rights reserved.

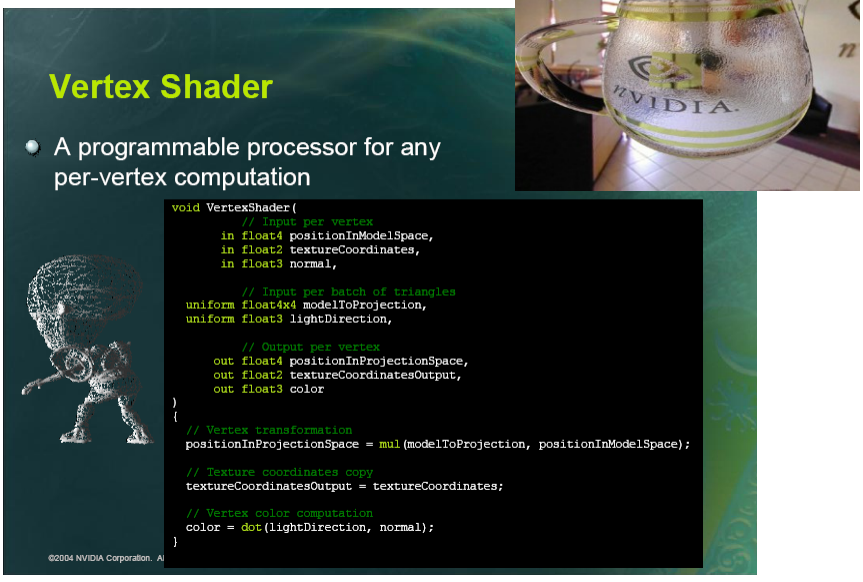

Department of Computer Science
'95-'98: Texture Mapping & Z-Buffer
CPU
Application / Geometry Stage
2D Triangles
Textures
System Memory
Bus (PCI)
GPU
Rasterization Stage
Rasterizer
Texture Unit
Raster Operations Unit
2D Triangles
Textures
Frame Buffer
Video Memory
©2004 NVIDIA Corporation. All rights reserved.


Department of Computer Science
Raster Operations Unit (ROP)
Rasterizer
Texture Unit
Fragments
Raster Operations Unit
Fragment
Screen Position (x, y)
Alpha Value a
Depth z
Color (r, g, b)
tested against scissor rectangle
tested against reference value
stencil buffer value at (x, y) tested against reference value
tested against z-buffer value at (x, y) (Visibility test)
blended with color buffer value at (x, y)
$K_{src} * Color_{src} + K_{dst} * Color_{src}$
(src = fragment, dst = color buffer)
Scissor Test
Alpha Test
Stencil Test
Z Test
Alpha Blending
Frame Buffer
Stencil Buffer
Z-Buffer
Color Buffer
Pixels
©2004 NVIDIA Corporation. All rights reserved.


Department of Computer Science
1999-2000: Transform and Lighting
CPU
Application Stage
3D Triangles
Textures
System Memory
Bus (AGP)
GPU
"Fixed Function Pipeline"
Geometry Stage
Transform & Lighting Unit
Rasterization Stage
Rasterizer
Texture Unit
Register Combiner
Raster Operations Unit
3D Triangles
Textures
Frame Buffer
Video Memory
©2004 NVIDIA Corporation. All rights reserved.


Wanda and Bubble
©2004 NVIDIA Corporation. All rights reserved.


2001: Programmable Vertex Shader
CPU
Application Stage
3D Triangles
Textures
System Memory
Bus (AGP)
GPU
Geometry Stage
Vertex Shader (no flow control)
Rasterization Stage
Rasterizer w/ Z-Cull
Texture Unit (4 Textures)
Register Combiner
Raster Operations Unit
3D Triangles
Textures
Frame Buffer
Video Memory
©2004 NVIDIA Corporation. All rights reserved.



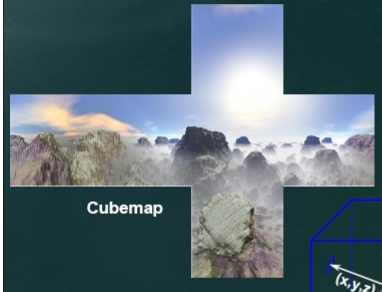
Need more realism!

Bump Mapping

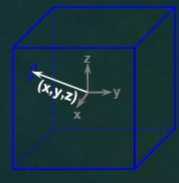
- Bump mapping involves fetching the per-pixel normal from a normal map texture (instead of using the interpolated vertex normal) in order to compute lighting at a given pixel

Diffuse light | Normal Map | Diffuse light with bumps

Cubic Texture Mapping



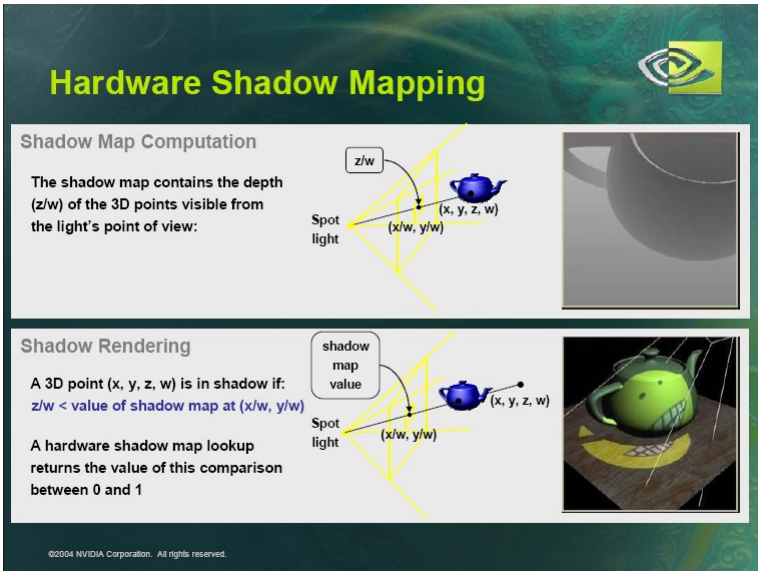
Cubemap



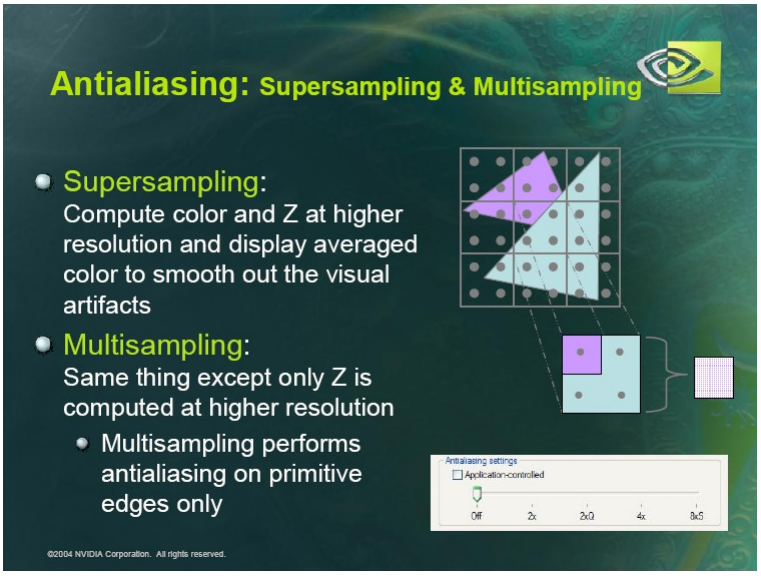
Cubemap lookup
in direction (x, y, z)

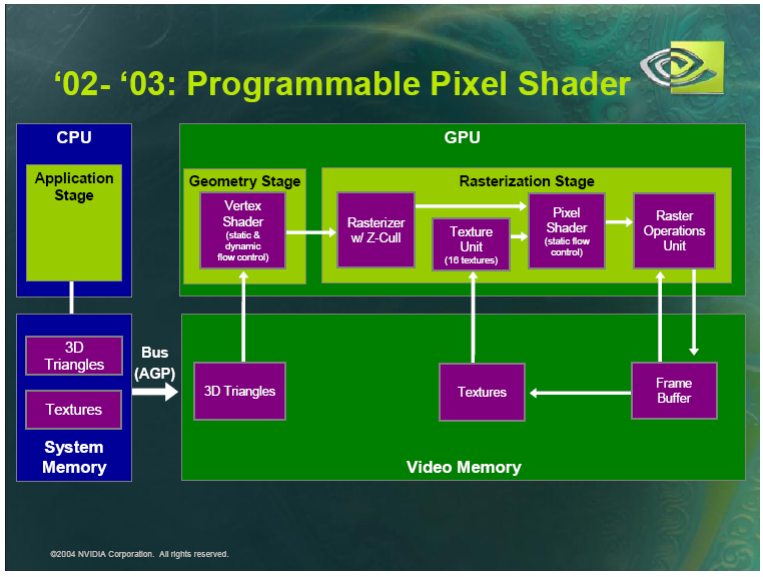
Environment Mapping
Reflection vector
is used to lookup
into the cubemap

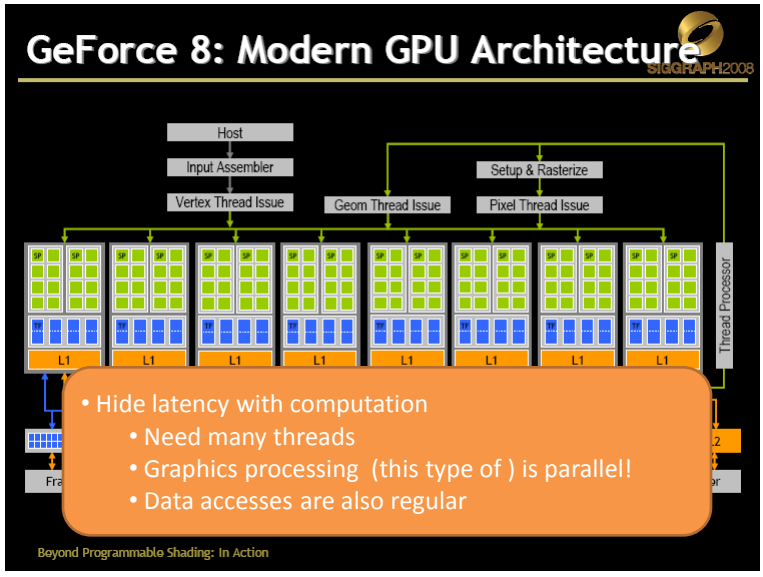
Hardware Shadow Mapping



Antialiasing: Supersampling & Multisampling




'02- '03: Programmable Pixel Shader
CPU
Application Stage
GPU
Geometry Stage
Vertex Shader
(static & dynamic flow control)
Rasterization Stage
Rasterizer w/ Z-Cull
Texture Unit
(16 textures)
Pixel Shader
(static flow control)
Raster Operations Unit
3D Triangles
Textures
System Memory
Bus (AGP)
3D Triangles
Textures
Frame Buffer
Video Memory
©2004 NVIDIA Corporation. All rights reserved.


GeForce 8: Modern GPU Architecture
SIGGRAPH2008
Host
Input Assembler
Vertex Thread Issue
Geom Thread Issue
Setup & Rasterize
Pixel Thread Issue
Thread Processor
L1
• Hide latency with computation
• Need many threads
• Graphics processing (this type of) is parallel!
• Data accesses are also regular
Beyond Programmable Shading: In Action

General purpose programmer's view