NVIDIA VRWORKS
Comprehensive SDK for VR Developers

GRAPHICS
LENSES MATCHED SHADING
SINGLE PASS STEREO
MULTIRES SHADING
VR SLI

HEADSET
CONTEXT PRIORITY
DIRECT MODE
FRONT BUFFER RENDERING

TOUCH & PHYSICS
PHYSX

PROFESSIONAL
WARP & BLEND
SYNCHRONIZATION
GPU AFFINITY

AUDIO
VRWORKS AUDIO

VIDEO
VRWORKS 360 VIDEO
GPUDIRECT FOR VIDEO
NVIDIA VRWORKS ADOPTION

**HEADSETS**

- VIVE
- oculus
- OSVR
- STARVR

**ENGINES**

- Unreal Engine
- Unity

**SOME PRO APPLICATIONS**

- Autodesk® VRED™
- FUnique
- Fuzor
- IC.IDO
- CAVRNUS
- SheenCity.com
- OPTIS
- Z Cam

3
NVIDIA VRWORKS
Comprehensive SDK for VR Developers

<table>
<thead>
<tr>
<th>GRAPHICS</th>
<th>HEADSET</th>
<th>TOUCH &amp; PHYSICS</th>
<th>AUDIO</th>
<th>VIDEO</th>
</tr>
</thead>
<tbody>
<tr>
<td>LENSMATCHED SHADING</td>
<td>CONTEXT PRIORITY</td>
<td>PHYSX</td>
<td>VRWORKS AUDIO</td>
<td>VRWORKS 360 VIDEO</td>
</tr>
<tr>
<td>SINGLE PASS STEREO</td>
<td>DIRECT MODE</td>
<td>PROFESSIONAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTireshading</td>
<td>FRONT BUFFER RENDERING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VR SLI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CAD DATA IN VR
GRAPHICS PIPELINE

VR Workloads

124M Pix/s
N vertices
60 Hz

457M Pix/s
2N vertices
90 Hz

3x
Application

Preprocessing

Geometric Pipeline

Rasterization Fragment Shader

Postprocessing

~3.6x
**NVIDIA VRWORKS**
Comprehensive SDK for VR Developers

<table>
<thead>
<tr>
<th>GRAPHICS</th>
<th>HEADSET</th>
<th>TOUCH &amp; PHYSICS</th>
<th>AUDIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lens Matched</td>
<td>Context Priority</td>
<td>PhysX</td>
<td>VRWORKS Audio</td>
</tr>
<tr>
<td>Shading</td>
<td></td>
<td>Professional</td>
<td></td>
</tr>
<tr>
<td>Single Pass</td>
<td>Direct Mode</td>
<td>Warp &amp; Blend</td>
<td></td>
</tr>
<tr>
<td>Stereo</td>
<td>Front Buffer</td>
<td>Synchronization</td>
<td></td>
</tr>
<tr>
<td>Multires Shading</td>
<td>Rendering</td>
<td>GPU Affinity</td>
<td></td>
</tr>
<tr>
<td>VR SLI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Video**

360 VRWorks Video

GPU Direct For Video
SINGLE PASS STEREO

Render eyes separately

Doubles CPU and GPU load

Left Eye (Pass 1)

Right Eye (Pass 2)
SINGLE PASS STEREO

Using SPS to improve rendering performance

Single Pass Stereo uses Simultaneous Multi-Projection architecture

Draw geometry only once

Vertex/Geometry stage runs once

Outputs two positions for left/right

Only rasterization is performed per-view
SINGLE PASS STEREO

OpenGL

In OpenGL via \texttt{GL_NV_stereo_view_rendering}

Create texture array for rendering left and right eye simultaneously

No other changes needed, shaders perform SPS
SINGLE PASS STEREO
OpenGL - Vertex Shader

Calculate projection space position

\[
\text{proj} \_\text{pos} = \text{proj} \times \text{view} \times \text{model} \times \text{inPosition};
\]

Output both positions via different builtin variables, only x component may differ

\[
\text{gl\_Position} = \text{proj} \_\text{pos} + \text{vec4}(\text{offset}, 0, 0, 0);
\]

\[
\text{gl\_SecondaryPositionNV} = \text{proj} \_\text{pos} - \text{vec4}(\text{offset}, 0, 0, 0);
\]

Use declaration and value of \text{gl\_Layer} to route output to layers 0 and 1 of tex array

\[
\text{layout(secondary_view_offset=1) out highp int gl\_Layer;}
\]

\[
\text{gl\_Layer} = 0;
\]
SINGLE PASS STEREO

Vulkan

In Vulkan via `VK_NVX_multiview_per_view_attributes`

Create layered texture image and view for rendering left and right simultaneously

Requires MultiView support

Update: `VK_KHX_multiview` ratified to `VK_KHR_multiview` in Vulkan 1.1
SINGLE PASS STEREO
Vulkan - Vertex Shader

Calculate projection space position

\[
\text{proj\_pos} = (\text{proj} * \text{view} * \text{model} * \text{inPosition}).xyz;
\]

Standard MultiView - specify once, may execute shader twice

\[
\text{gl\_Position} = \text{proj\_pos} + \text{UBO.offsets}[\text{gl\_ViewIndex}];
\]

With per-view attributes - also specify positions explicitly, execute shader only once

\[
\begin{align*}
\text{gl\_PositionPerViewNV}[0] &= \text{proj\_pos} + \text{UBO.offsets}[0]; \\
\text{gl\_PositionPerViewNV}[1] &= \text{proj\_pos} + \text{UBO.offsets}[1];
\end{align*}
\]
Single Pass Stereo: Benefits in geometry bound scenarios

Heavy fragment shaders will reduce scaling

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>MultiView</th>
<th>MultiView with per-view attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat shading</td>
<td>7.1</td>
<td>6.7</td>
<td>3.7</td>
</tr>
<tr>
<td>+ Phong</td>
<td>7.2</td>
<td>6.8</td>
<td>4.5</td>
</tr>
<tr>
<td>+ Noise</td>
<td>7.2</td>
<td>6.9</td>
<td>4.9</td>
</tr>
</tbody>
</table>

NVIDIA Quadro P6000, Scene with 17.6M faces, frame times in ms
NVIDIA VRWORKS
Comprehensive SDK for VR Developers

<table>
<thead>
<tr>
<th>GRAPHICS</th>
<th>HEADSET</th>
<th>TOUCH &amp; PHYSICS</th>
<th>AUDIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>LENS MATCHED SHADING</td>
<td>CONTEXT PRIORITY</td>
<td>PHYSX</td>
<td>VRWORKS AUDIO</td>
</tr>
<tr>
<td>SINGLE PASS STEREO</td>
<td>DIRECT MODE</td>
<td>PROFESSIONAL</td>
<td>VIDEO</td>
</tr>
<tr>
<td>MULTires SHADING</td>
<td>FRONT BUFFER RENDERING</td>
<td>WARP &amp; BLENDE</td>
<td>VRWORKS 360 VIDEO</td>
</tr>
<tr>
<td>VR SLI</td>
<td></td>
<td>SYNCHRONIZATION</td>
<td>GPUDIRECT FOR VIDEO</td>
</tr>
</tbody>
</table>
VR SLI
Crash course

Left view data

Geometry
Materials

Right view data

L  R  R
VR SLI
Scaling 1 vs 2 GPUs

Scaling = \frac{2 \times \text{GPU}}{\text{GPU} + \text{Copy}}
Scaling determined by workload and copy time

VR SLI

Scaling = \frac{2 \times \text{GPU}}{\text{GPU} + \text{Copy}}

Typical render resolution for Vive
1512 x 1680 (per eye)

Copy time over PCIe (@6GB/s)
1.5ms

Max scaling with 11ms frame time
\frac{2 \times 9.5ms}{9.5ms + 1.5ms} = 1.72

\text{Scaling factor}

\text{Workload (ms)}
VR SLI

Higher resolutions limit scalability

\[ Scaling = \frac{2 \times GPU}{GPU + \text{Copy}} \]

Vive Pro render resolution

2016 x 2240

Copy time over PCIe (@6GB/s)

2.8ms

Max scaling with 11ms frame time

\[ \frac{2 \times 8.2ms}{8.2ms + 2.8ms} = 1.49 \]
Copy times can hurt scaling with higher resolutions

NVLink on dual Quadro GP100: **4x faster** than PCIe 3.0

Copy time for Vive Pro (2016 x 2240): 0.7ms

Max scaling with 11ms frame time

\[
\frac{2 \times 10.3\text{ms}}{10.3\text{ms} + 0.7\text{ms}} = 1.87
\]

NVLink is used automatically if present

NVLink speed measured with 2 bridges, copy via OpenGL multicast, single frame of HTC Vive, on HP z840 workstation
VR SLI
Upcoming HMDs can improve scaling

Some upcoming HMDs have one display cable per eye.

SLI system: Plug each cable into one GPU.

Eliminate inter-GPU copies by presenting on each GPU.

Needs support from the VR runtime: near future.

Working on 4-GPU configuration:
- Two pairs of GPUs connected via NVLink.
- One Multicast context spanning the configuration.
- Split Frame Rendering (SFR), NVLink copies.

HMD image courtesy of Starbreeze.
OPENGL & VULKAN
OPENGL MULTICAST 2
Feedback on Multicast led to new functionality

Command & data broadcast
BufferSubData to specific GPU
CopyImageSubData & CopyBufferSubData
GPU-GPU Framebuffer Blit
Global barrier & directed sync functions
GPU Masks
Per-GPU sample locations
Per-GPU queries

Dynamic Multicast toggle (WGL_NV_multigpu_context)
GPU_ID built-in in GLSL shader
Per-GPU viewports & scissors
Texture & Buffer upload mask
Asynchronous copies
New extension WGL_NV_multigpu_context: Request SLI mode per context

No need to restart application

Possible to share resources between contexts
New extension WGL_NV_multigpu_context: Request SLI mode per context

No need to restart application

Possible to share resources between contexts

On toggle:

Clean up per-GPU resources

Keep scene data

Alternate Frame Rendering (AFR)
MULTICAST 2

GPU ID built-in

Multicast v1 required per-GPU uploads
Larger code changes in some renderers
Add shader built-in
  Upload all views to all GPUs
  Use per-GPU data in shaders
Renderer can remain unchanged
Just modify shaders instead
MULTICAST 2
Per-GPU Viewports & Scissors

Add new function to set viewports and scissors per GPU

Per-GPU Lens Matched Shading
MULTICAST 2
Per-GPU Viewports & Scissors

Add new function to set viewports and scissors per GPU

Per-GPU Lens Matched Shading
Per-GPU Multi Resolution Shading
MULTICAST 2
Per-GPU Viewports & Scissors

Add new function to set viewports and scissors per GPU

Per-GPU Lens Matched Shading
Per-GPU Multi Resolution Shading
Easily set up Split Frame Rendering (SFR)
Multicast provides per-GPU buffer uploads

Asymmetrical functionality wrt texture upload functions

Add new mask function to modify texture & buffer uploads

```
glUploadGpuMaskNVX( GLbitfield mask );
```

Useful for simpler per-GPU texture streaming

Conserve PCIe bandwidth
MULTICAST 2
Asynchronous Copies

Multicast copies stall source GPU while copy takes place
Easy to use because of implicit synchronization
New copy functions do not stall, but also need more synchronization
   `glAsyncCopyBufferSubDataNV(…)`
   `glAsyncCopyImageSubDataNV(…)`
Copy while both GPUs can continue rendering
Allows for more complex rendering algorithms
MULTICAST 2
Asynchronous Copies - Use case

Render shadow maps (SM)
Start async copies of SMs to other GPU
Render z-prepass per GPU & eye
Wait for copy to finish
Render output images
VR SLI
Vulkan

Update: `VK_KHX_device_group` ratified to `VK_KHR_device_group` with Vulkan 1.1

Make sure to use the right extension/Vulkan version combination!

Usage is the same, so migration is painless
VR SLI

Recap

VR SLI covers a wide variety of workloads

Almost perfect load balancing between left/right eye and two GPUs

Copy overhead and view independent workloads limit scaling

NVLink can help improve scaling

Dual-input HMDs can eliminate copy overhead
NVIDIA VRWORKS
Comprehensive SDK for VR Developers

GRAPHICS
- Lens Matched Shading
- Single Pass Stereo
- MultiRes Shading
- VR SLI

HEADSET
- Context Priority
- Direct Mode
- Front Buffer Rendering

TOUCH & PHYSICS
- PHYSX
- Professional
- Warp & Blend
- Synchronization
- GPU Affinity

AUDIO
- VRWORKS Audio

VIDEO
- VRWORKS 360 Video
- GPU Direct for Video
HMD OPTICS
Countering Lens Distortion
HMD RENDERING
Oversampling near the borders

Rendered Image

Displayed Image
LENS MATCHED SHADING

Four Viewports

Original Image

LMS Image
In OpenGL via `GL_NV_clip_space_w_scaling` extension

Set up four viewports, rendering full resolution

Set scissors to each quadrant

```c
glScissorArray(0, 4, scissors);
```

W scaling parameters

```c
glViewportPositionWScaleNV(i, Wx, Wy);
```
LENS MATCHED SHADING
Vulkan

In Vulkan via `VK_NV_clip_space_w_scaling` extension
Set up four viewports, rendering full resolution
Set scissors to each quadrant
`VkPipelineViewportWScalingStateCreateInfoNV`
W scaling parameters:
Use the viewport struct / set on creation
Dynamic state & `vkCmdSetViewportWScalingNV`
LENS MATCHED SHADING

Extreme example, Wx = 2.0  Wy = 2.0
LMS can improve performance of Raster / Fragment stage

Trade-off between quality and performance
# NVIDIA VRWORKS

Comprehensive SDK for VR Developers

<table>
<thead>
<tr>
<th>GRAPHICS</th>
<th>HEADSET</th>
<th>TOUCH &amp; PHYSICS</th>
<th>AUDIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>LENSMATCHED SHADING</td>
<td>CONTEXT PRIORITY</td>
<td>PHYSX</td>
<td>VRWORKS AUDIO</td>
</tr>
<tr>
<td>SINGLE PASS STEREO</td>
<td>DIRECT MODE</td>
<td>WARPA &amp; BLEND</td>
<td></td>
</tr>
<tr>
<td>MULTIPASS SHADING</td>
<td>FRONT BUFFER RENDERING</td>
<td>SYNCHRONIZATION</td>
<td></td>
</tr>
<tr>
<td>VR SLI</td>
<td></td>
<td>GPU AFFINITY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VIDEO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>360</td>
<td>VRWORKS 360 VIDEO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GPUDIRECT FOR VIDEO</td>
</tr>
</tbody>
</table>
KHRONOS
Cross-Platform VR Standard
OPEN XR
Version 1.0 vision

Application

VR run-time A
  HMD A

VR run-time B
  HMD B

VR run-time C
  Mobile HMD
OPEN XR
Long term vision

Application
- VR run-time A
  - AR goggles
- VR run-time B
  - HMD
- VR run-time C
  - CAVE
MORE INFO @ GTC

More talks @ GTC 2018

https://developer.nvidia.com/vrworks-gtc

https://developer.nvidia.com/designworks-gtc

Connect with the Experts

CE8141 - VR: GL, DX & VK - Mon, March 26 2-3PM

CE8116 - VRWorks - Tue, March 27 4-5PM
TRY IT OUT!
..and more information

NVIDIA VRWorks SDK provides OpenGL, Direct3D & Vulkan samples

developer.nvidia.com/vrworks

More detail in our previous GTC talks:

2017 - S7191 - Vulkan Technology Update
2016 - S6338 - VR Multi GPU Acceleration Featuring Autodesk VRED
2015 - S5668 - VR Direct: How NVIDIA Technology Is Improving The VR Experience
THANK YOU!