

# Accelerating Your VR Games with VRWorks

Manuel Kraemer



# Talk Overview

- NVIDIA Pascal Overview
- VRWorks Graphics Features
  - Multi-Res Shading, Lens Matched Shading
  - Single Pass Stereo, VRSLI
- SMP Assist (new)
- Vulkan extensions (new)
- VR Tools - Nsight, FCAT VR

# NVIDIA In VR



# NVIDIA VR

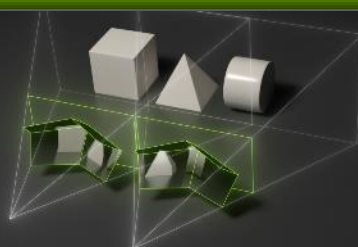
Powerful Hardware & Tools to Enhance Your VR Experiences

Hardware



SDKs & Tools

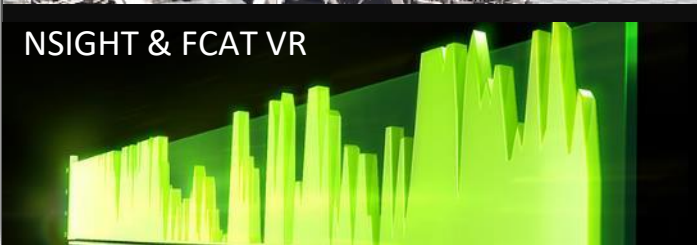
VRWorks



PhysX



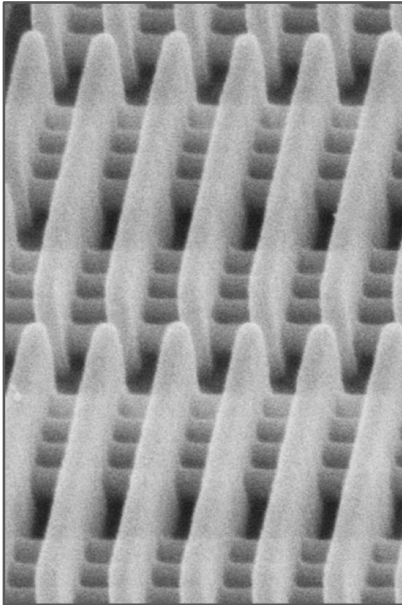
NSIGHT & FCAT VR



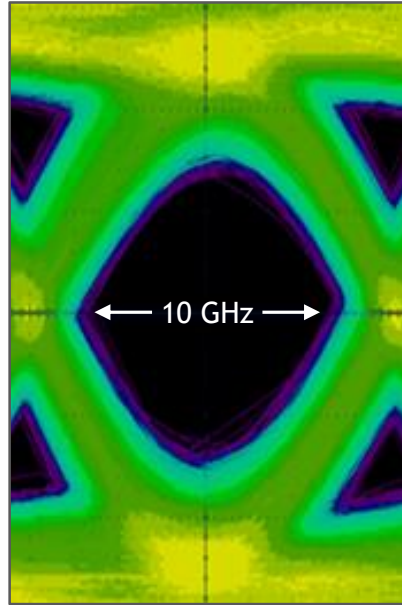
Applications



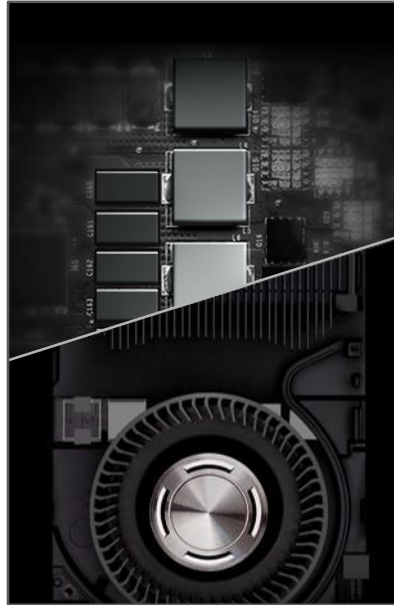
# NVIDIA Pascal GPU Architecture



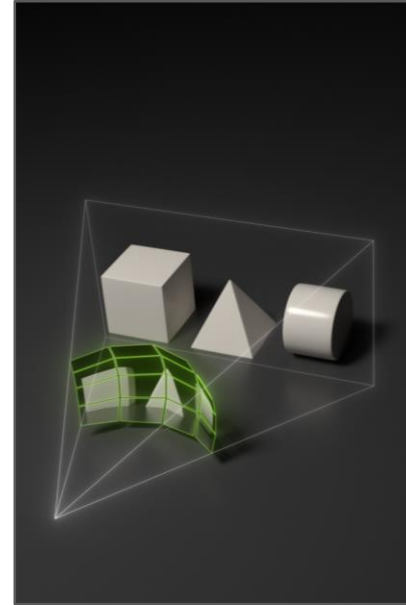
16NM FF



G5X



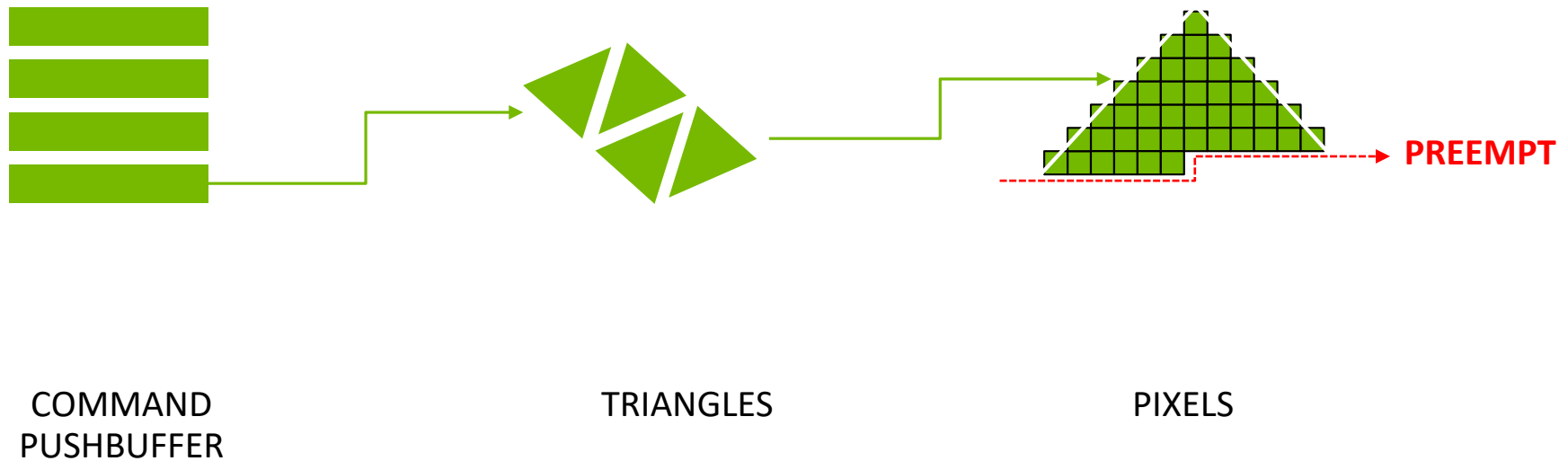
CRAFTSMANSHIP



SIMULTANEOUS MULTI-PROJECTION  
& PRE-EMPTION

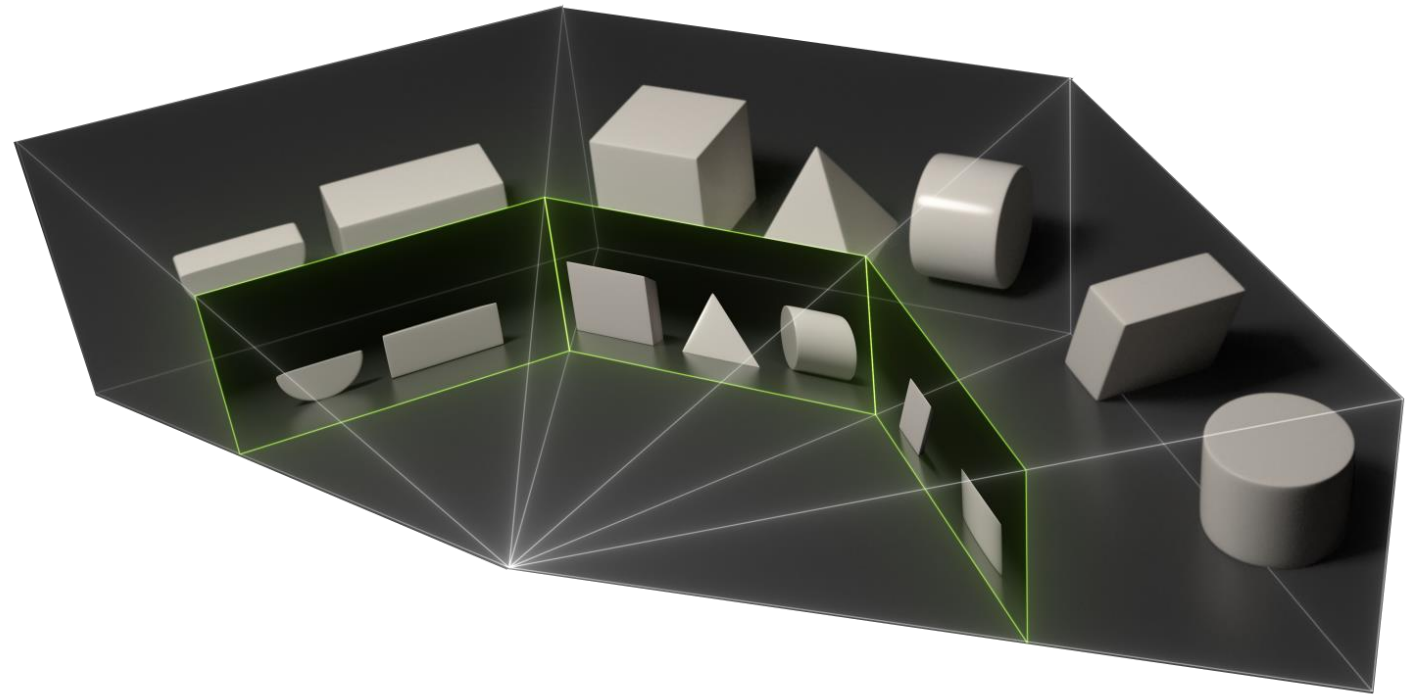
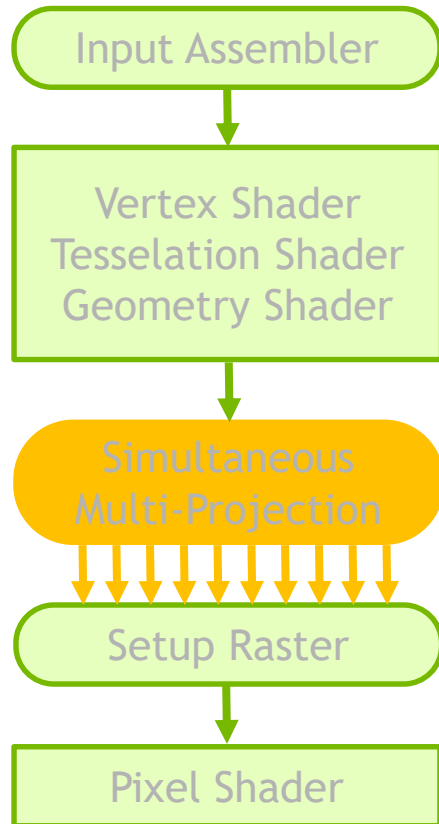
# NVIDIA PASCAL

- Pixel Level Preemption Improves Responsiveness For VR



# NVIDIA PASCAL

- Simultaneous Multi-Projection Engine



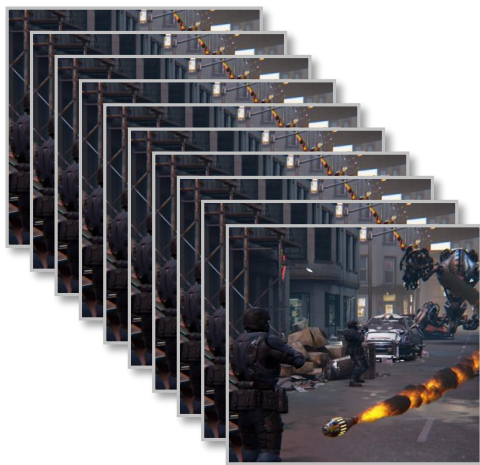
# VR GRAPHICS CHALLENGES



# VR Demands Serious Performance

Frame Rate

90FPS / 11 ms



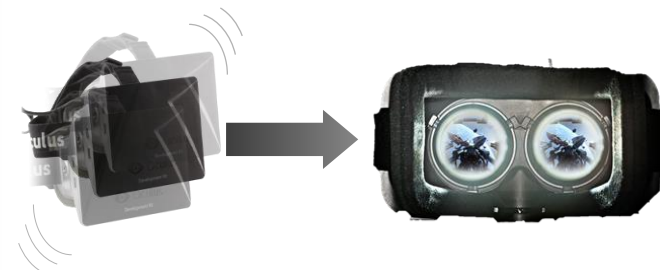
Resolution

800M Pixels/Sec

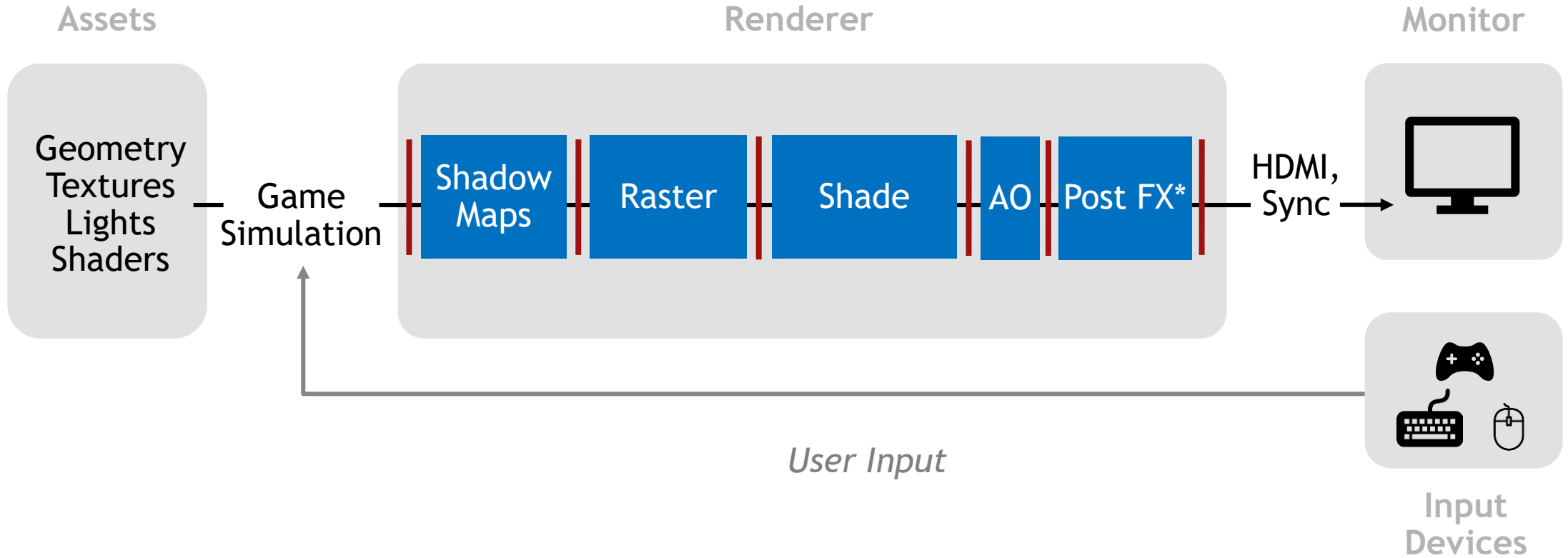


Latency

<20ms

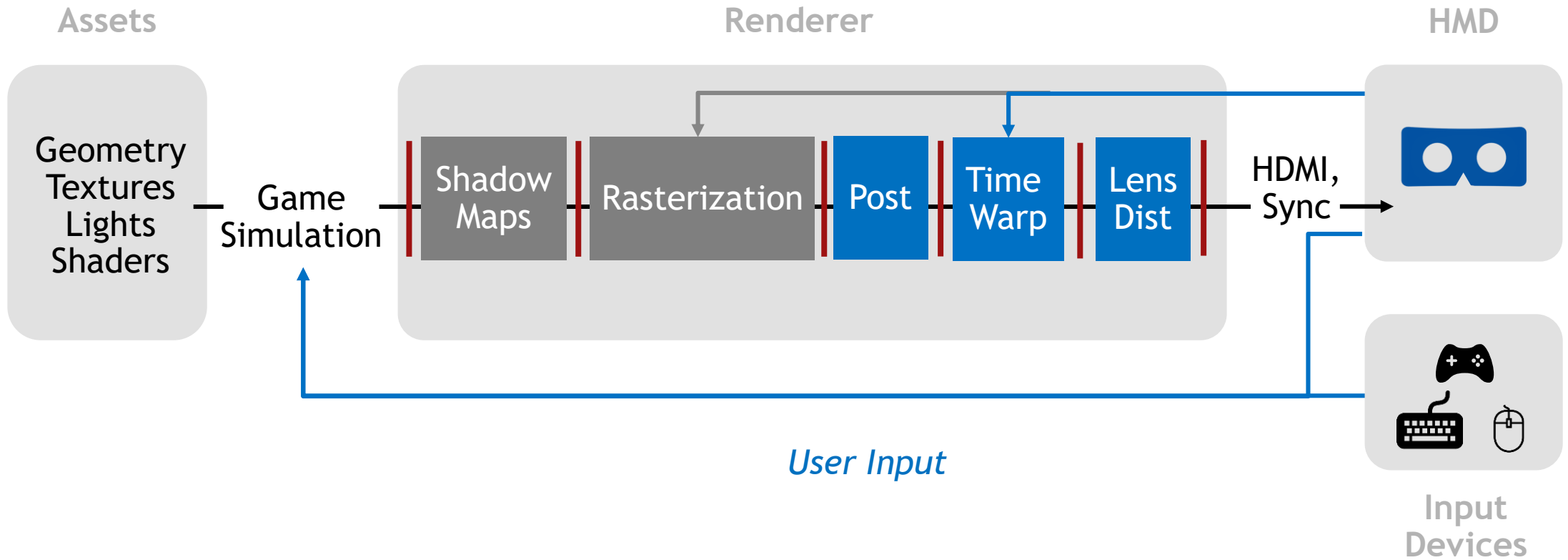


# 3D Game System

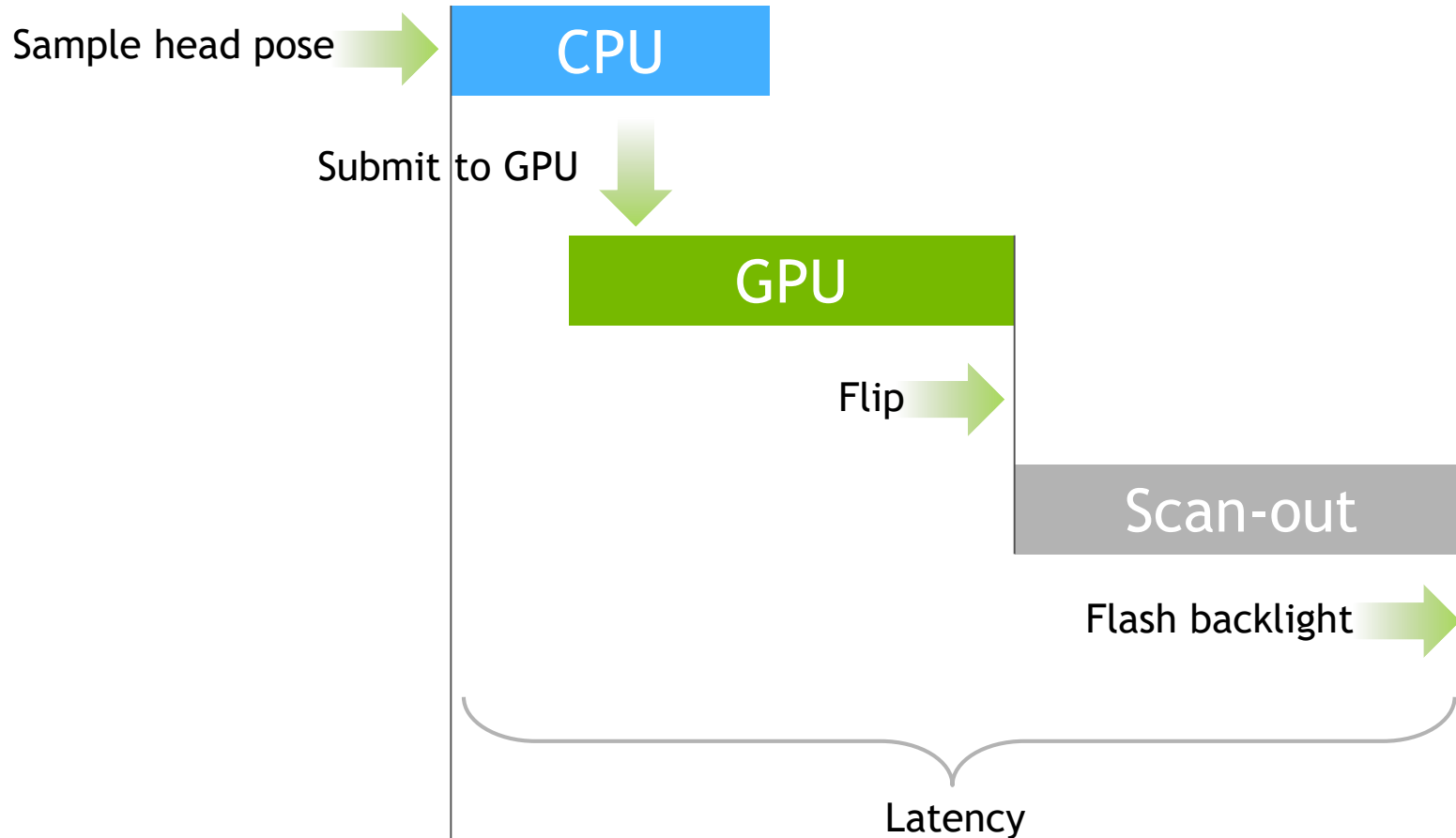


\* Includes depth of field, reflections, fog, color grading, motion blur, antialiasing

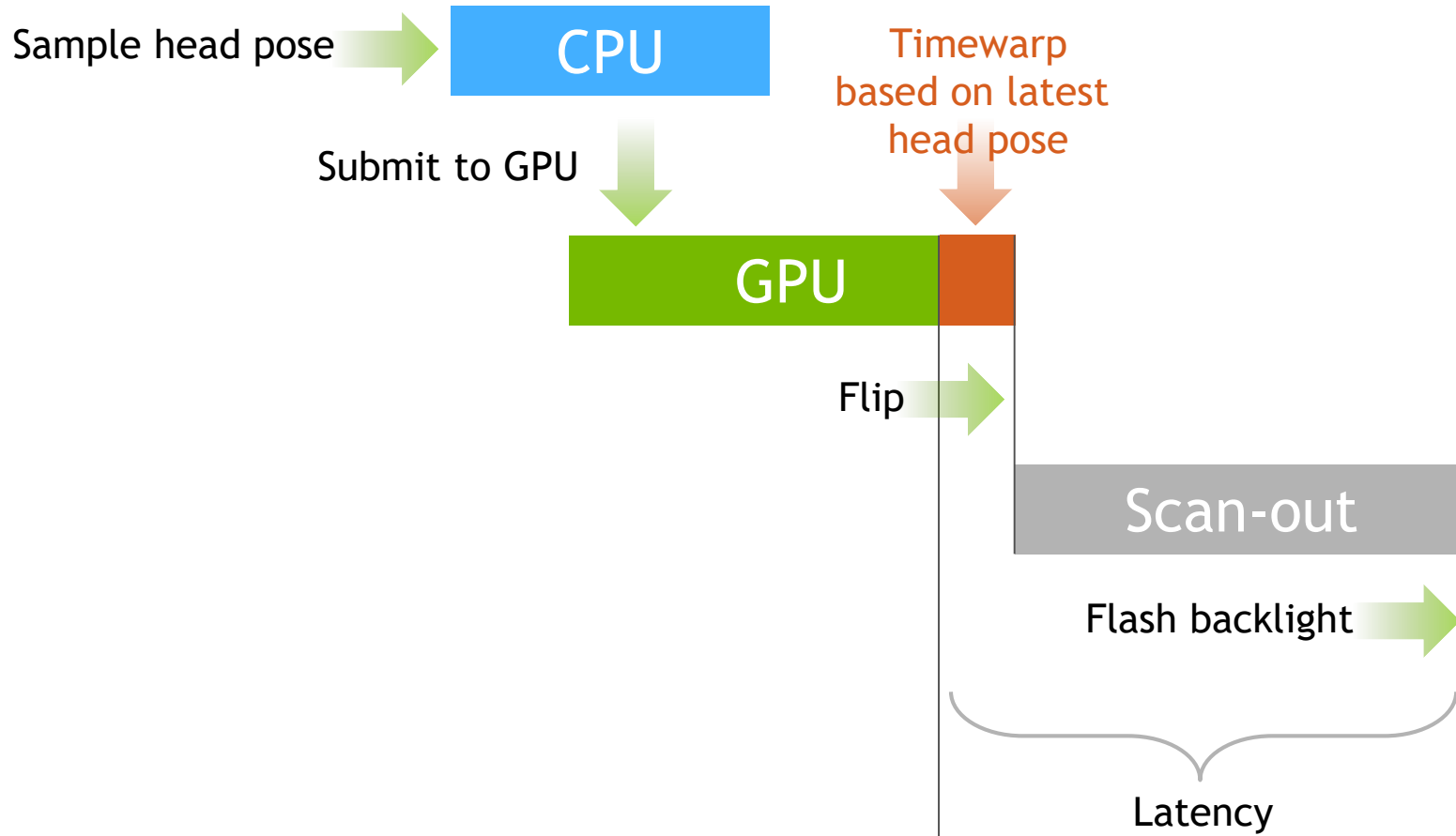
# VR Game System



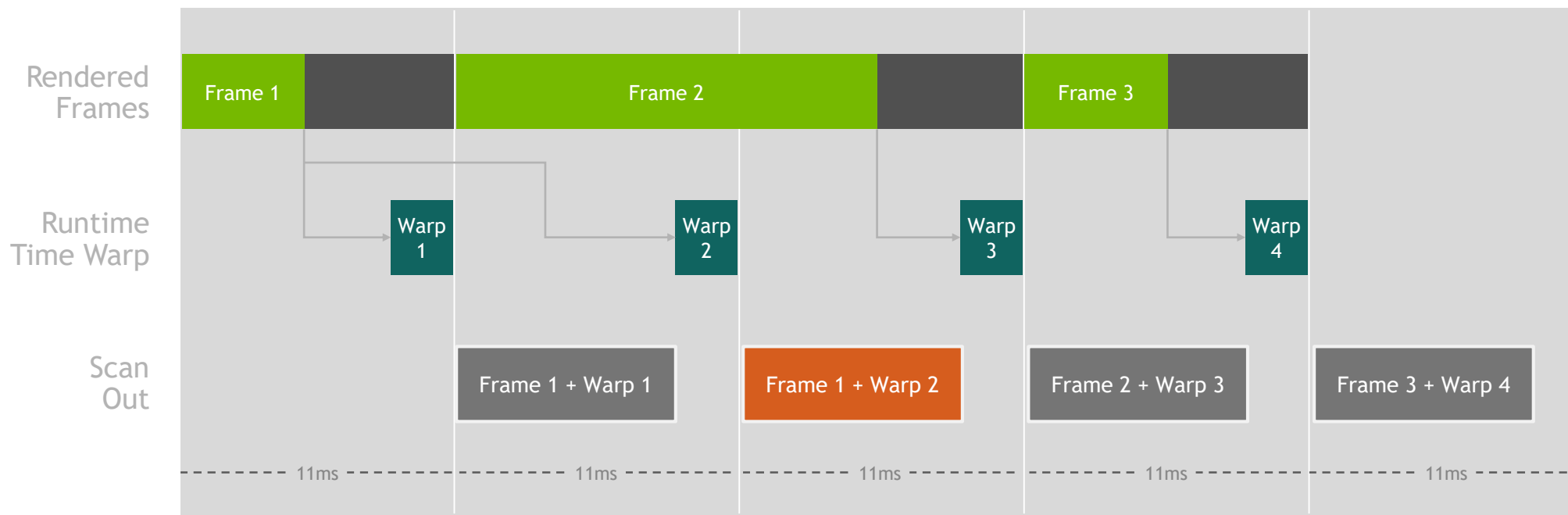
# VR LATENCY WITHOUT TIMEWARP



# VR LATENCY WITH TIMEWARP



# DROPPED FRAME



# Lens Distortion

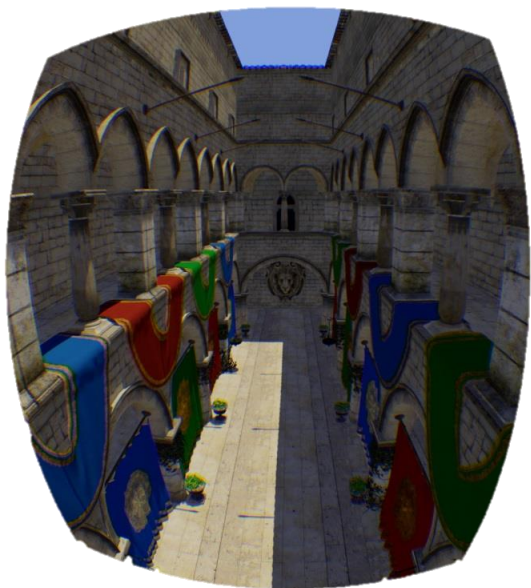
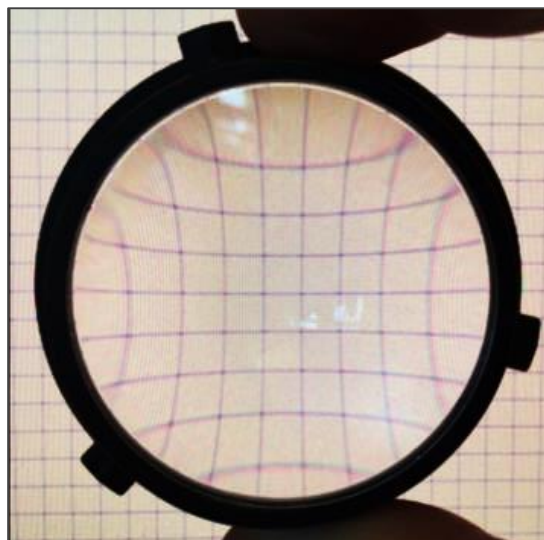
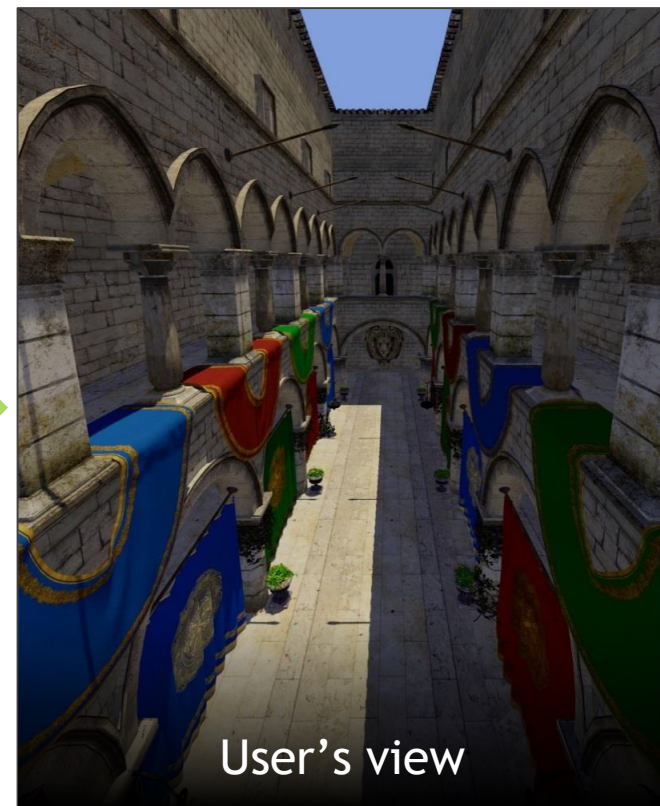


Image Displayed



Optics



User's view

# VRWorks



# NVIDIA VRWORKS

Bringing Reality to VR

## 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**



**GPUDIRECT  
FOR VIDEO**

# NVIDIA VRWORKS

Bringing Reality to VR

## GRAPHICS



LENS MATCHED  
SHADING



SINGLE PASS  
STEREO



MULTIRES  
SHADING



VR SLI

## HEADSET



CONTEXT  
PRIORITY



DIRECT  
MODE



FRONT BUFFER  
RENDERING

## TOUCH & PHYSICS



PHYSX

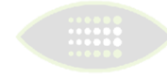
## PROFESSIONAL



WARP &  
BLEND



SYNCHRONIZATION



GPU  
AFFINITY

## AUDIO



VRWORKS  
AUDIO

## VIDEO



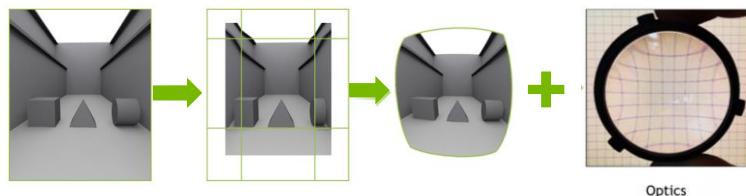
VRWORKS  
360 VIDEO



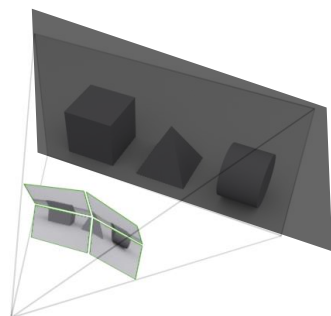
GPUDIRECT  
FOR VIDEO

# VRWORKS GRAPHICS

## RENDER LESS PIXELS

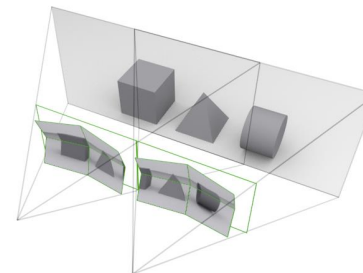


Multi-Resolution Shading (MRS)

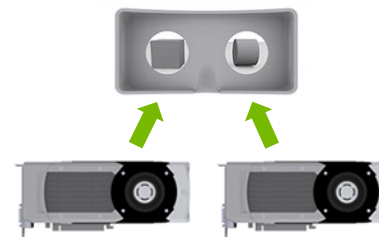


Lens Matched Shading (LMS)

## HANDLE LARGER SCENES



Single Pass Stereo (SPS)

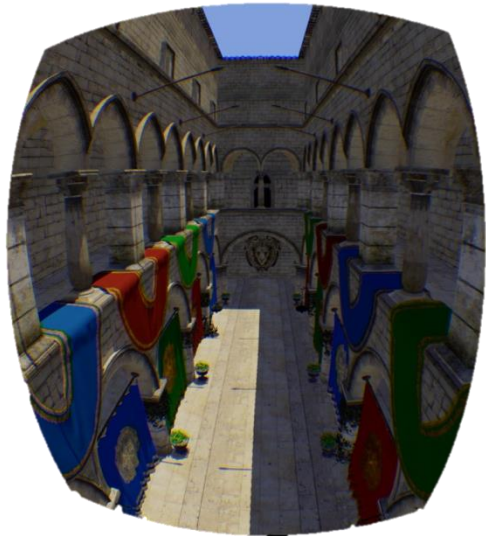


VRSLI

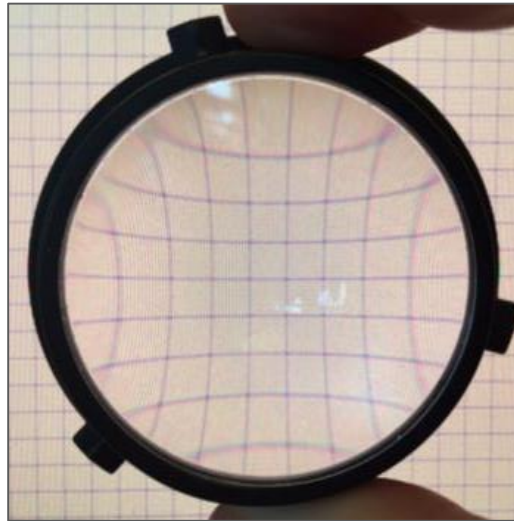


Render Less Pixels

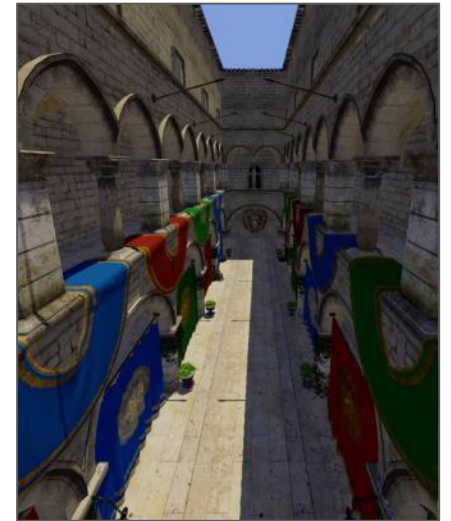
# VR OPTICS



LCD display

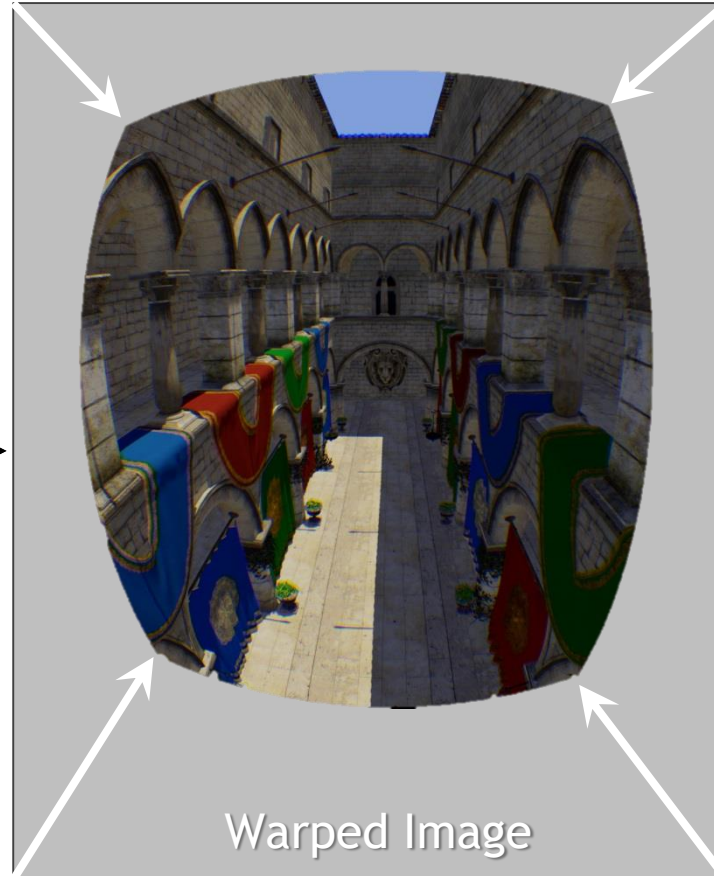


Optics



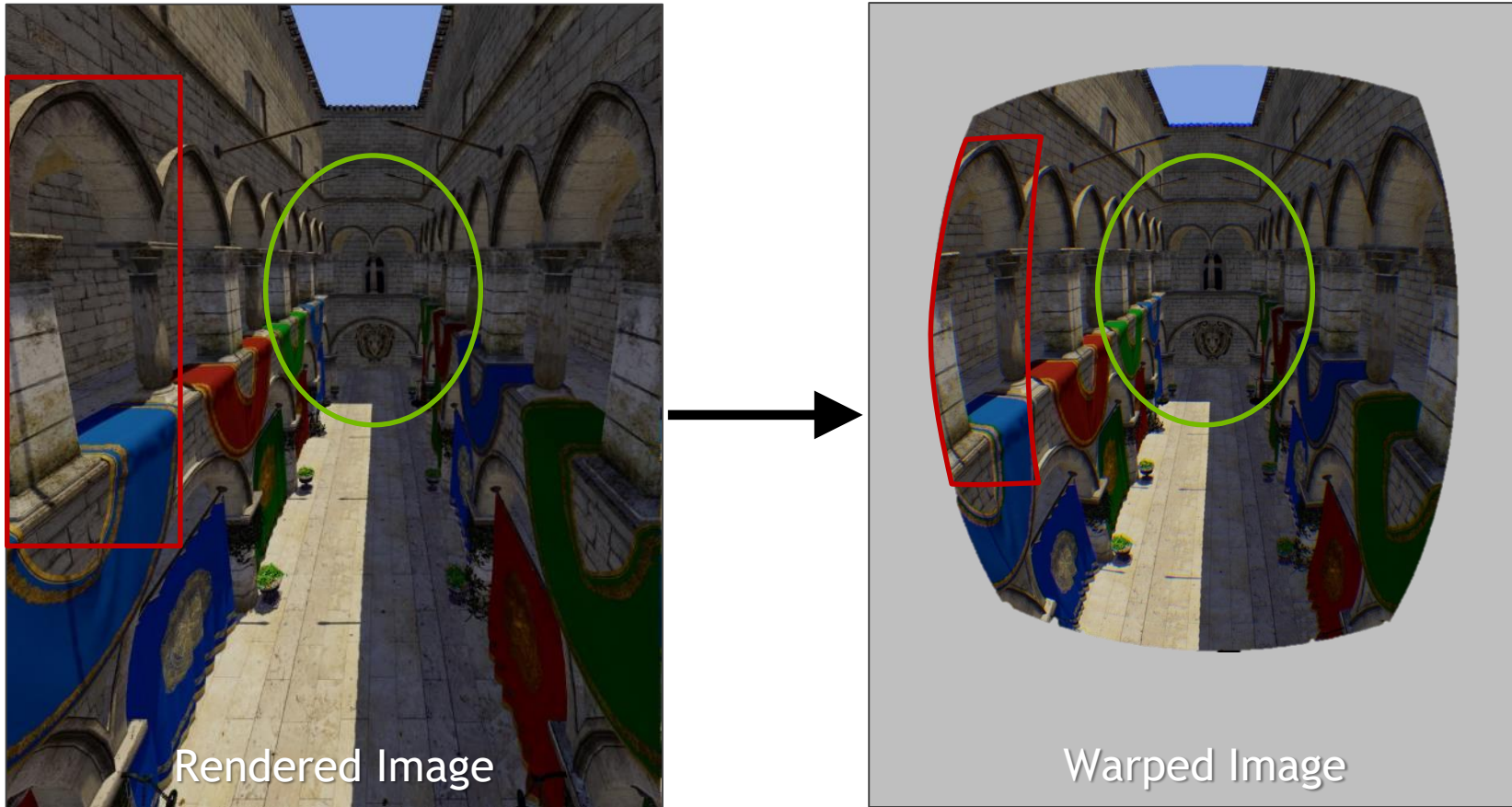
User's view

# VR RENDERING

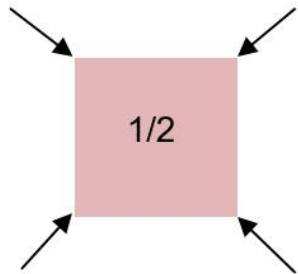
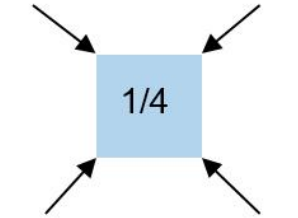
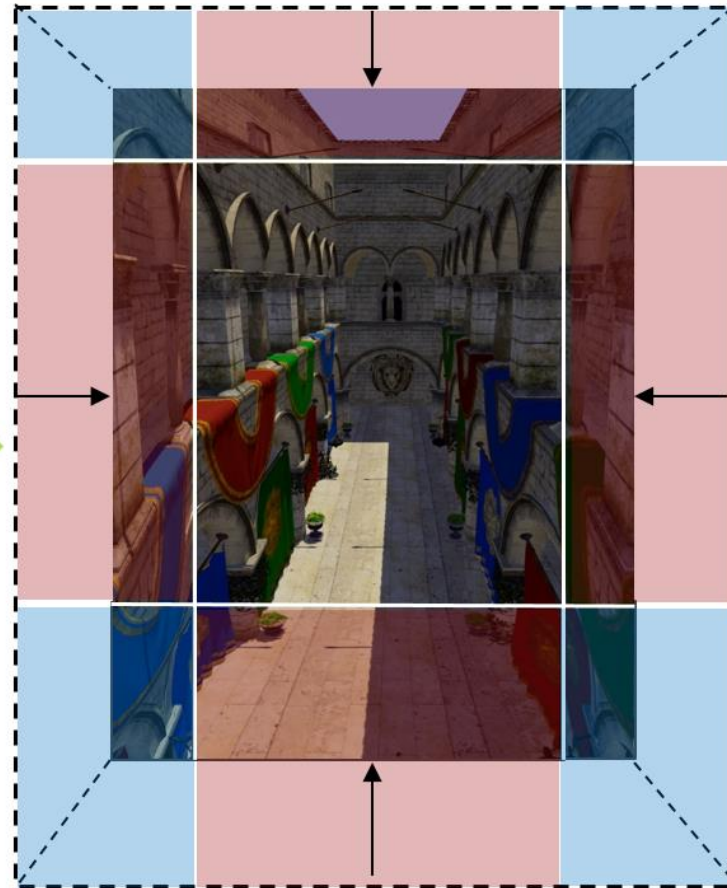
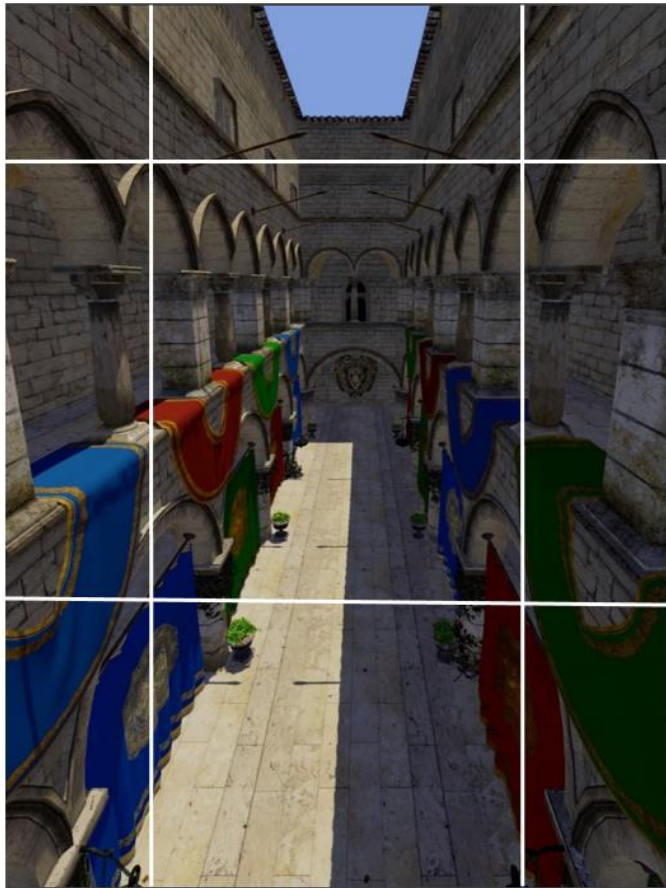


# VR RENDERING

GPU renders many pixels that never make it to screen



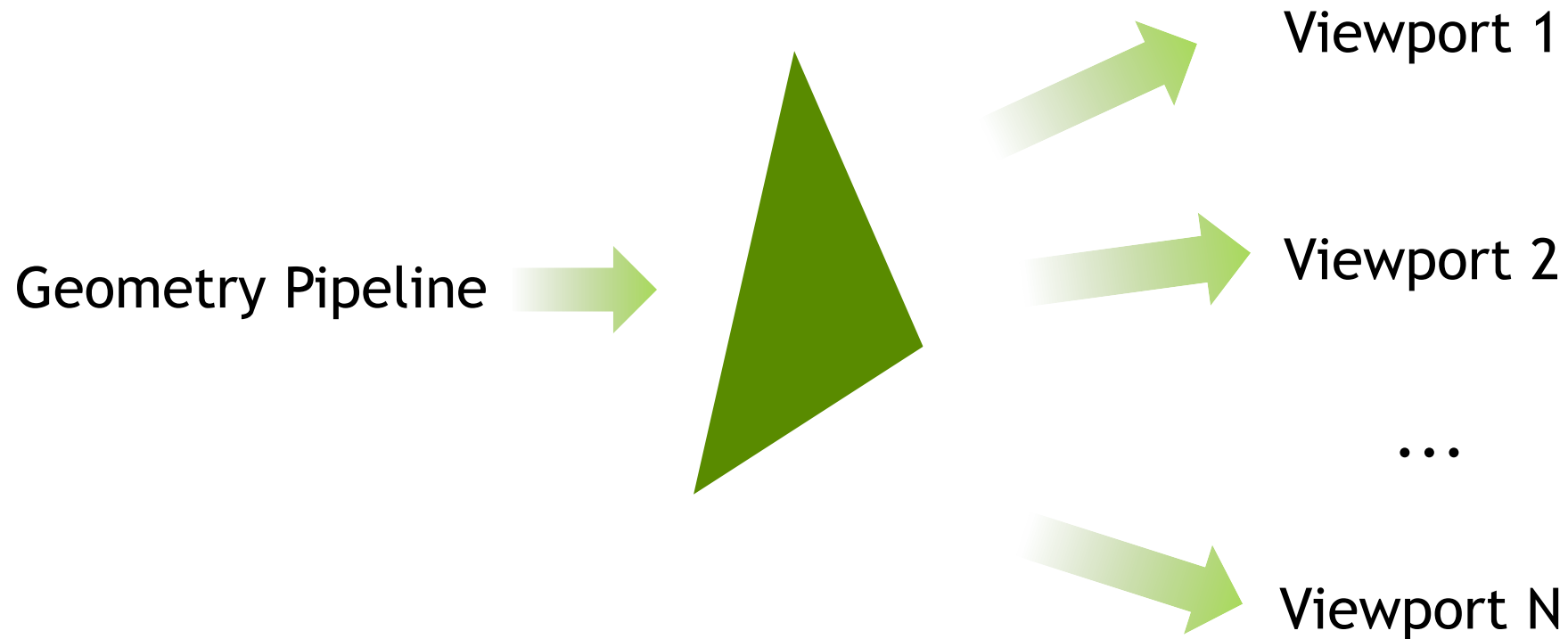
# VRWORKS MULTI-RES SHADING





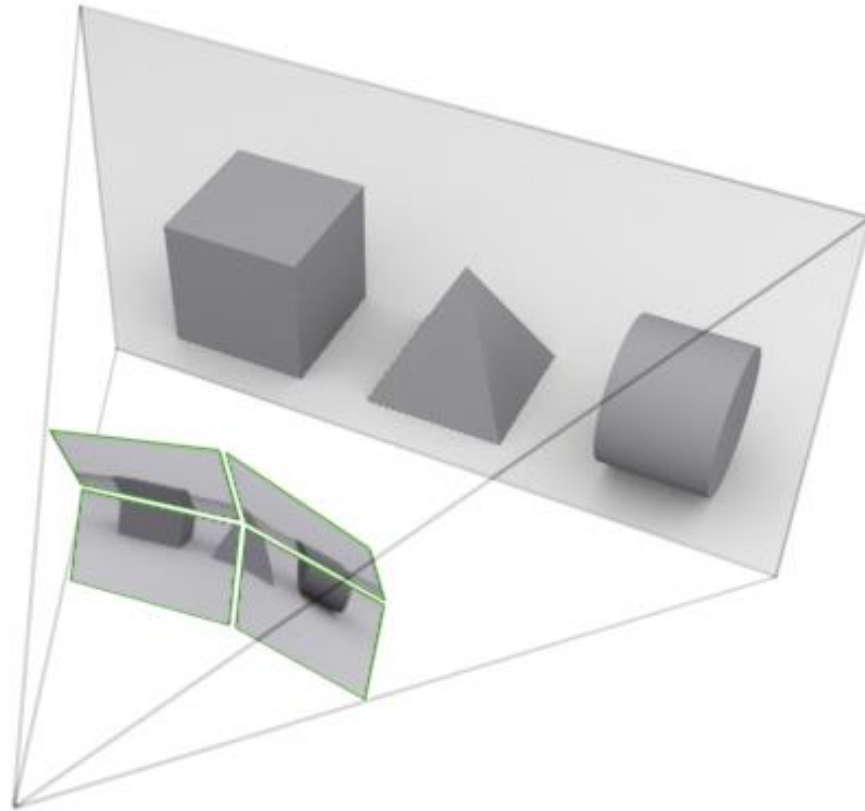
# Multi-resolution shading

Fast viewport broadcast on NVIDIA Maxwell and beyond GPUs



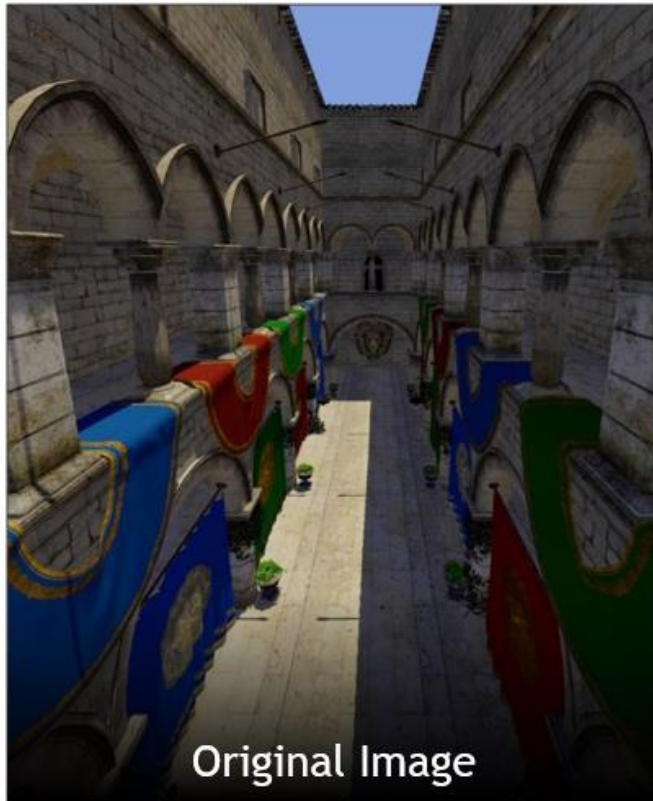
# VRWORKS LENS MATCHED SHADING

Renders to a lens corrected surface



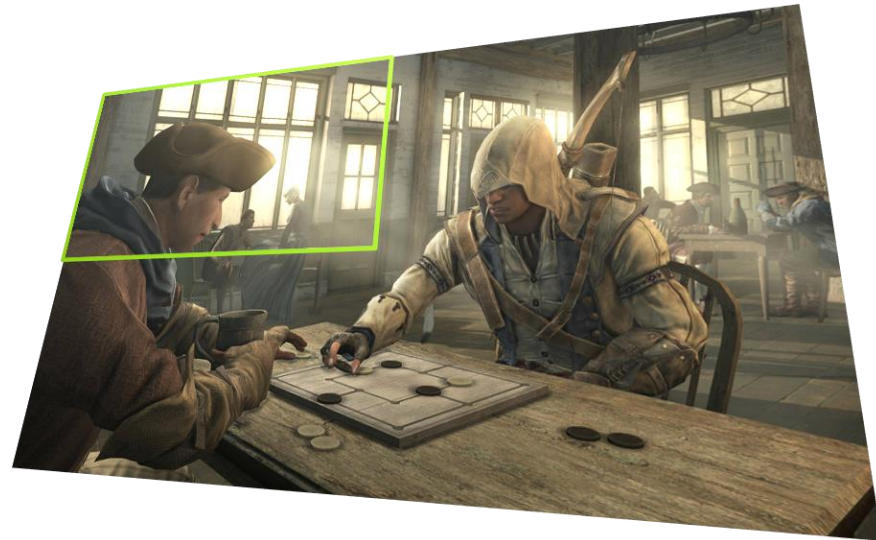
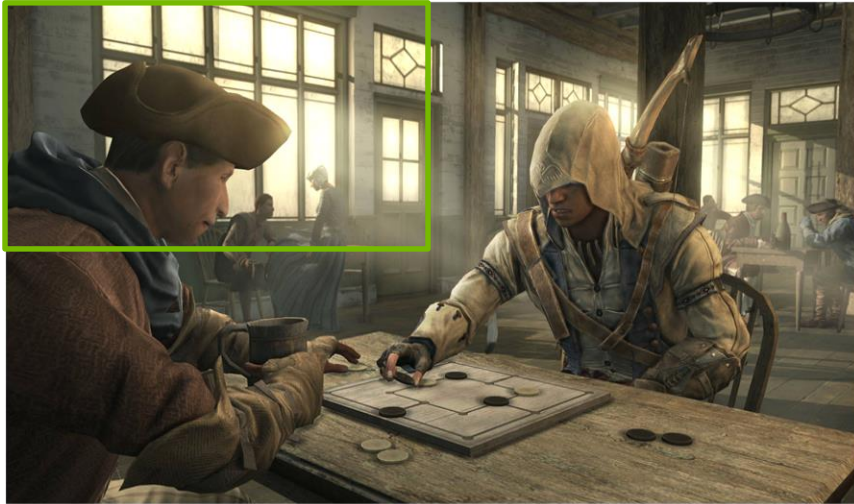
# LENS MATCHED SHADING

Renders to a lens corrected surface



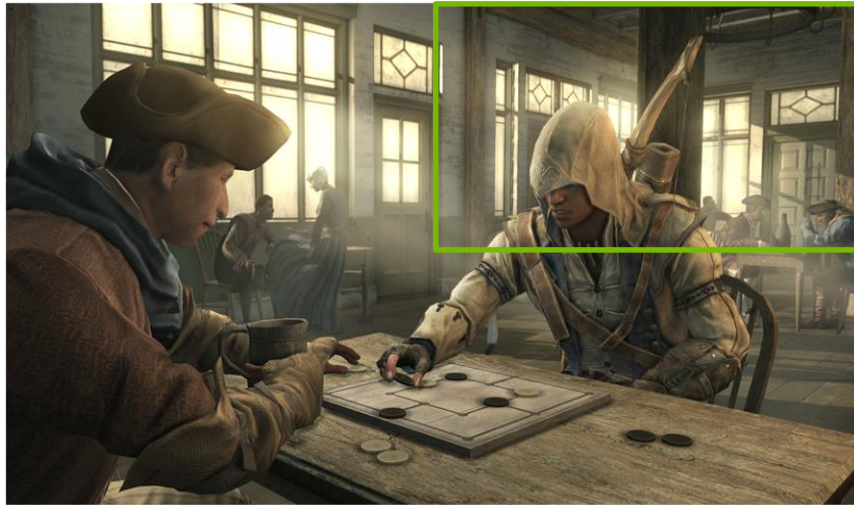
# LENS MATCHED SHADING

## Breakdown



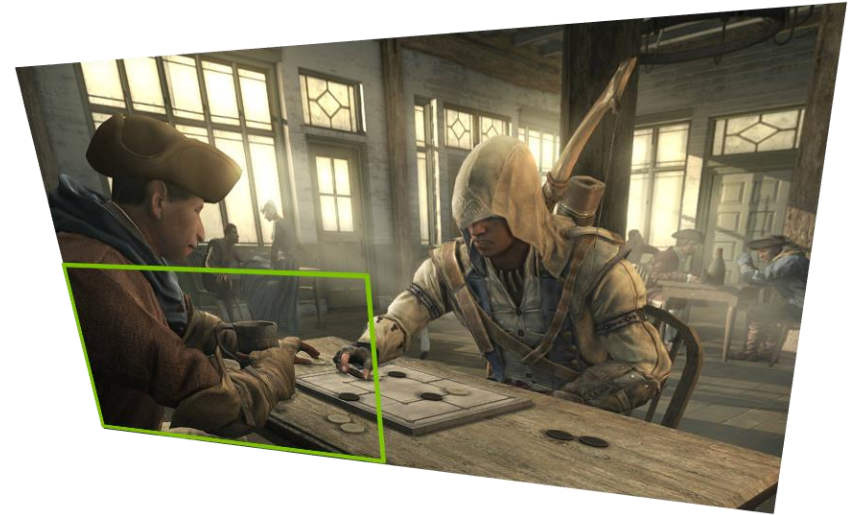
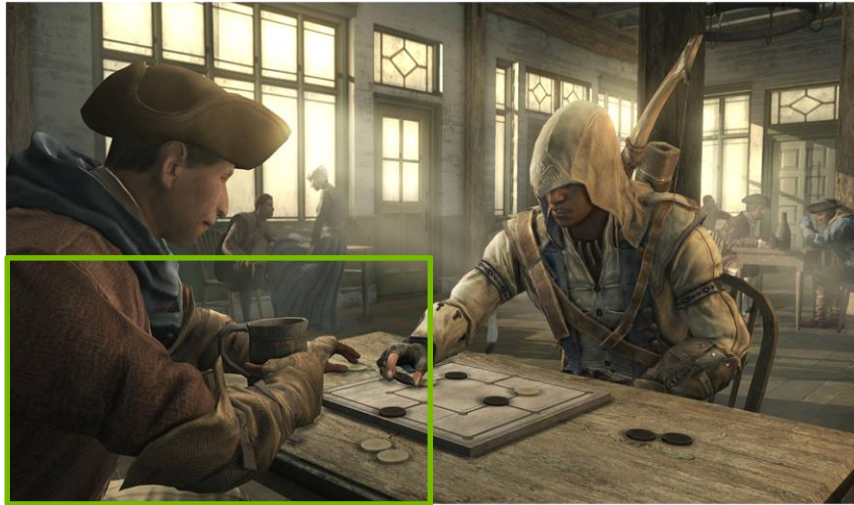
# LENS MATCHED SHADING

## Breakdown



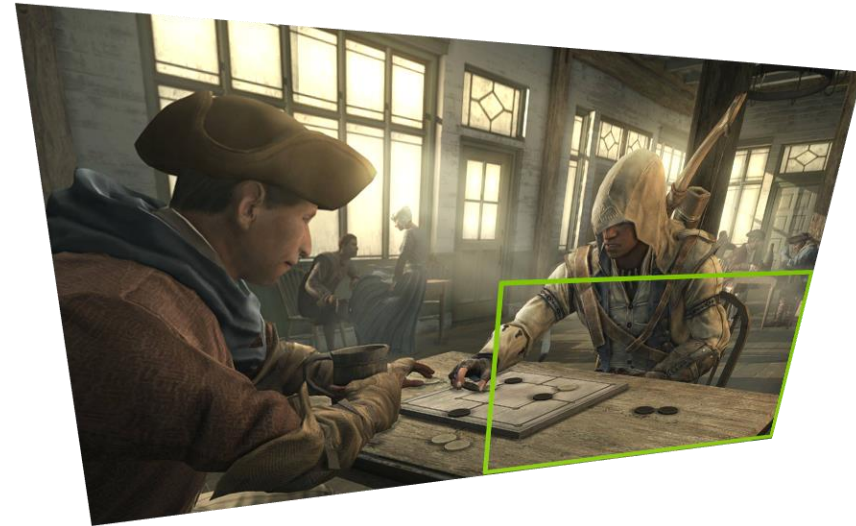
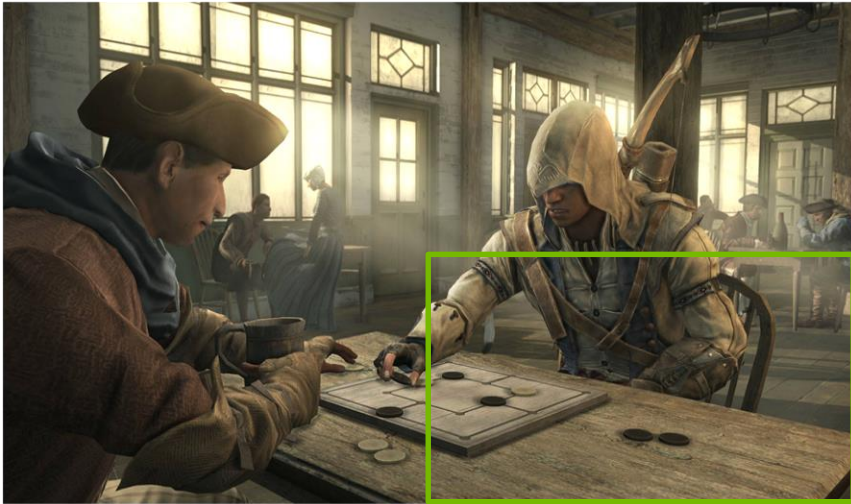
# LENS MATCHED SHADING

## Breakdown



# LENS MATCHED SHADING

## Breakdown



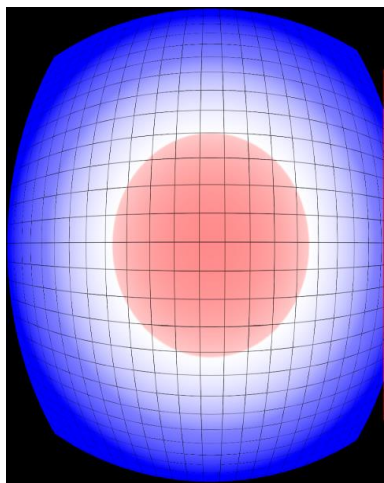
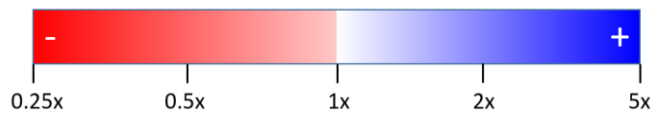
# LENS MATCHED SHADING

## Breakdown



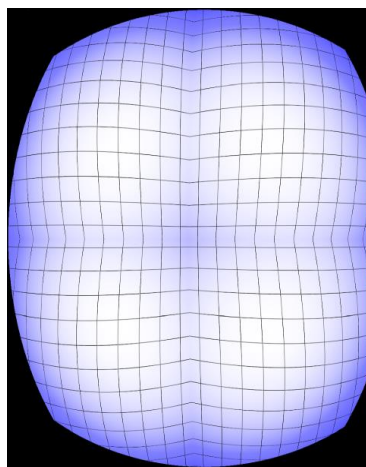


# LMS vs. MRS



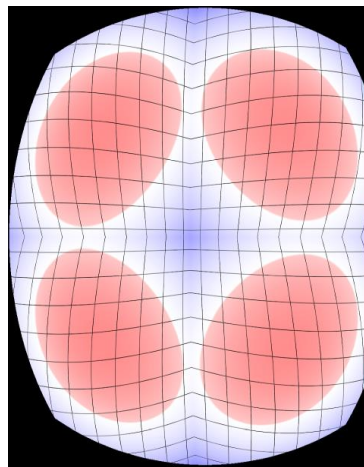
Baseline  
(no warp)  
2.54 MPix / eye

Quality  
(no undersampling)



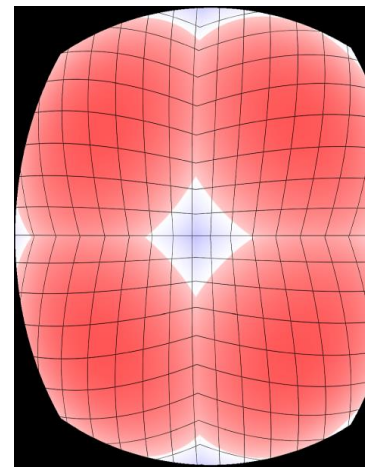
1.57 MPix / eye

Conservative  
(no worse than baseline)



1.17 MPix / eye

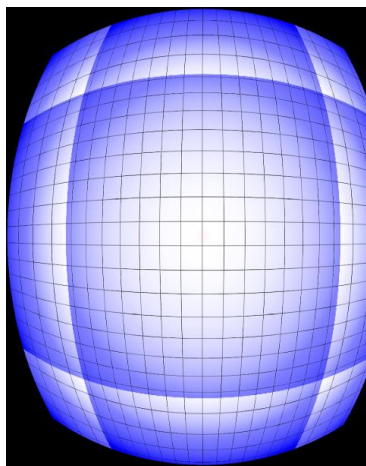
Aggressive  
(3/4 Reso. of conservative)



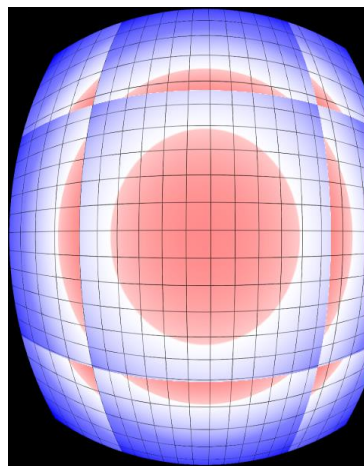
0.87 MPix / eye

LMS

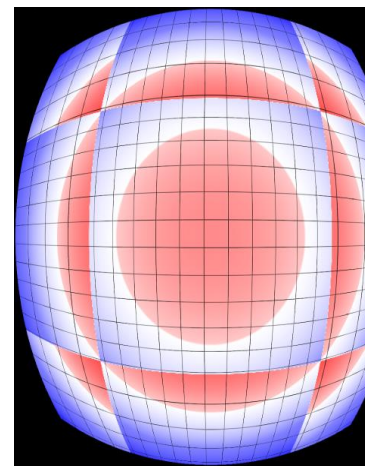
MRS



2.03 MPix / eye



1.58 MPix / eye



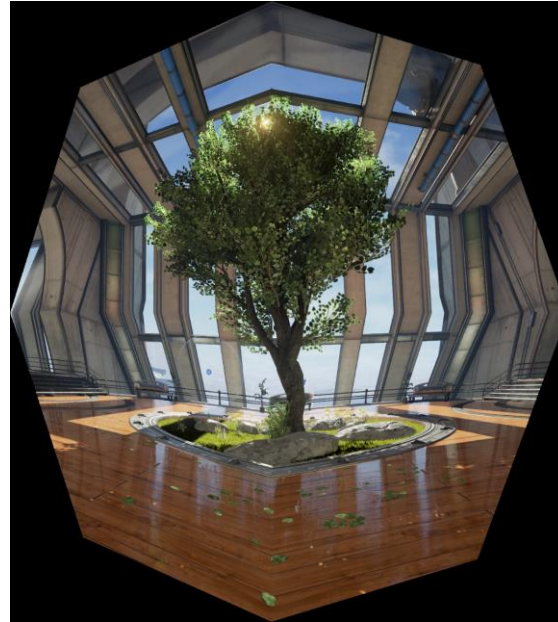
1.40 MPix / eye

# LMS / MRS Challenges

- Require unwarping
  - Minor speed and quality degradation
- Require application changes for
  - Setting / creating new “fast” geometry shaders
  - Set viewport / scissor state
  - Modifying shaders
  - Introducing SMP Assist to help with some of this

# Unwarping

- Oculus PC SDK 1.19 introduces native LMS support in the compositor!
- Avoids having to do it in-engine
- Improves quality and performance



# Introducing SMP Assist

## Helping with app complexity

- Application
  - Creates ID3DNvSMPAssist interface
  - Sets up projections
  - Calls Enable/Disable around render passes/draw calls
  - Use GetConstants results in shaders
- Driver
  - Creates & binds Fast Geometry Shaders for culling & projecting
  - Sets scissor and viewport rectangles
  - Returns constant buffer data needed

```
Interface ID3DNvSMPAssist
{
void Enable(IUnknown *pDevContext,
EyeIndex)

void Disable(IUnknown *pDevContext);

void GetConstants(...);

void SetupProjections(IUnknown *pDevice,);

void UpdateInstancedStereoData(IUnknown
*pDevice,...);
};
```

# SMP Assist levels of support

- **NV\_SMP\_ASSIST\_LEVEL\_FULL**
  - App selects a **pre-baked** MRS/LMS config (HMD type, quality level).
  - Driver handles correct setting of viewport, scissors and FastGS.
  - Driver provides constant buffer data for remapping.
- **NV\_SMP\_ASSIST\_LEVEL\_PARTIAL**
  - App provides a **custom** MRS/LMS config.
  - Driver handles correct setting of viewport, scissors and FastGS.
  - Driver provides constant buffer data for remapping.
- **NV\_SMP\_ASSIST\_LEVEL\_MINIMAL**
  - App provides viewports and scissors.
  - App sets FastGS as required.
  - App sets LMS params as required (NvAPI\_D3D\_SetModifiedWMode).
  - Driver handles correct setting of viewports and scissors.
  - Driver provides constant buffer for remapping.

# Shader Modification Example

```
/**
 * Pixel shader for rendering a directional light using a full screen quad.
 */
void DirectionalPixelMain(
    float2 InUV : TEXCOORD0,
    float3 ScreenVector : TEXCOORD1,
    float4 SVPos : SV_POSITION,
    out float4 OutColor : SV_Target0
)
{
    OutColor = 0;

    if (VRProjectionIsActive())
    {
        float4 LinearSvPos = SvPositionToLinearSvPosition(SVPos);
        ScreenVector = mul(float4(LinearSvPos.xy, 1, 1), View.SVPositionToTranslatedWorld).xyz;
    }

    float3 CameraVector = normalize(ScreenVector);

    FScreenSpaceData ScreenSpaceData = GetScreenSpaceData(InUV);

    // Only light pixels marked as using deferred shading
    BRANCH if( ScreenSpaceData.GBuffer.ShadingModelID > 0
#ifdef USE_LIGHTING_CHANNELS
        && (GetLightingChannelMask(InUV) & DeferredLightUniforms.LightingChannelMask)
#endif
    )
    {
        float SceneDepth = CalcSceneDepth(InUV);
        float3 WorldPosition = ScreenVector * SceneDepth + View.WorldCameraOrigin;

        FDeferredLightData LightData = SetupLightDataForStandardDeferred();

        uint2 Random = ScrambleTEA( uint2( SVPos.xy ) );
        Random.x ^= View.Random;
        Random.y ^= View.Random;

        OutColor = GetDynamicLighting(WorldPosition, CameraVector, ScreenSpaceData.GBuffer,
            ScreenSpaceData.AmbientOcclusion,
            ScreenSpaceData.GBuffer.ShadingModelID,
            LightData, GetPerPixelLightAttenuation(InUV), Random);
    }
}
```

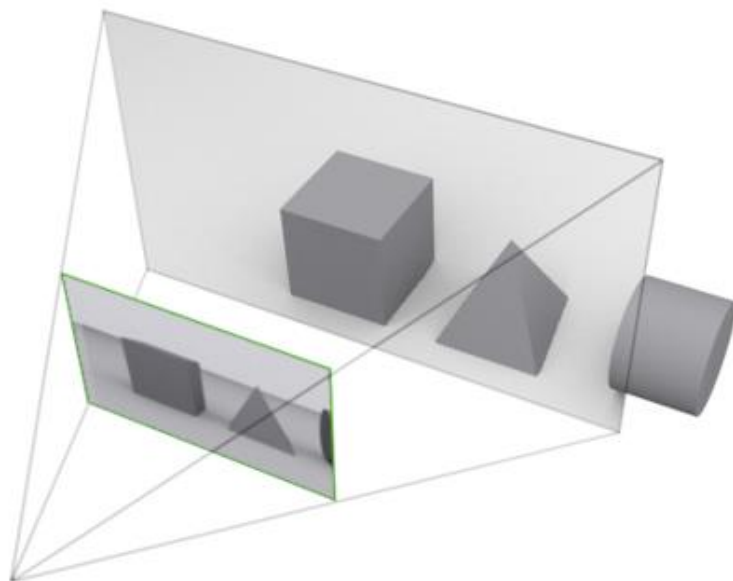
The input SVPos is in LMS space.  
So convert it to linear space, since CameraVector is used to calculate lighting with GBuffer data, which is also in linear space.

InUV is LMS space.  
When fetching data from GBuffers, use LMS space coordinates directly : GBuffer is indexed in LMS space.

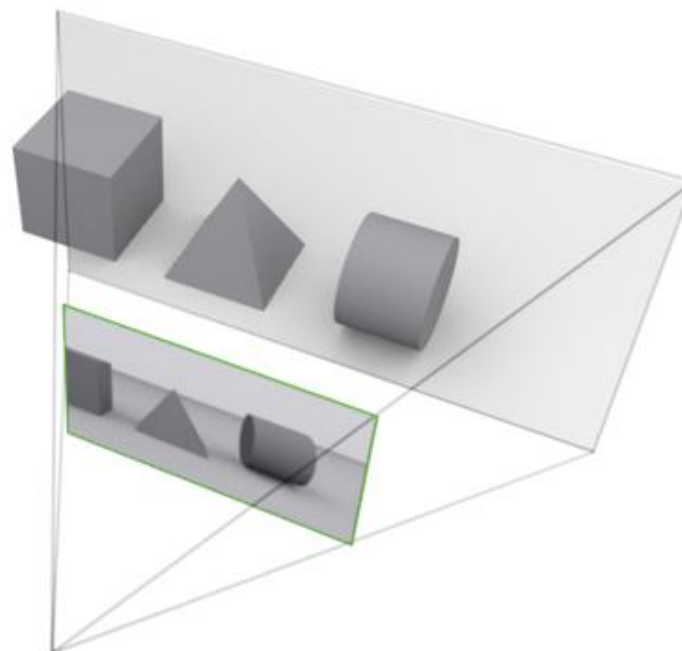
# Handle Larger Scenes

# TRADITIONAL STEREO RENDERING

Requires 2 geometry passes



Left Eye (Pass 1)

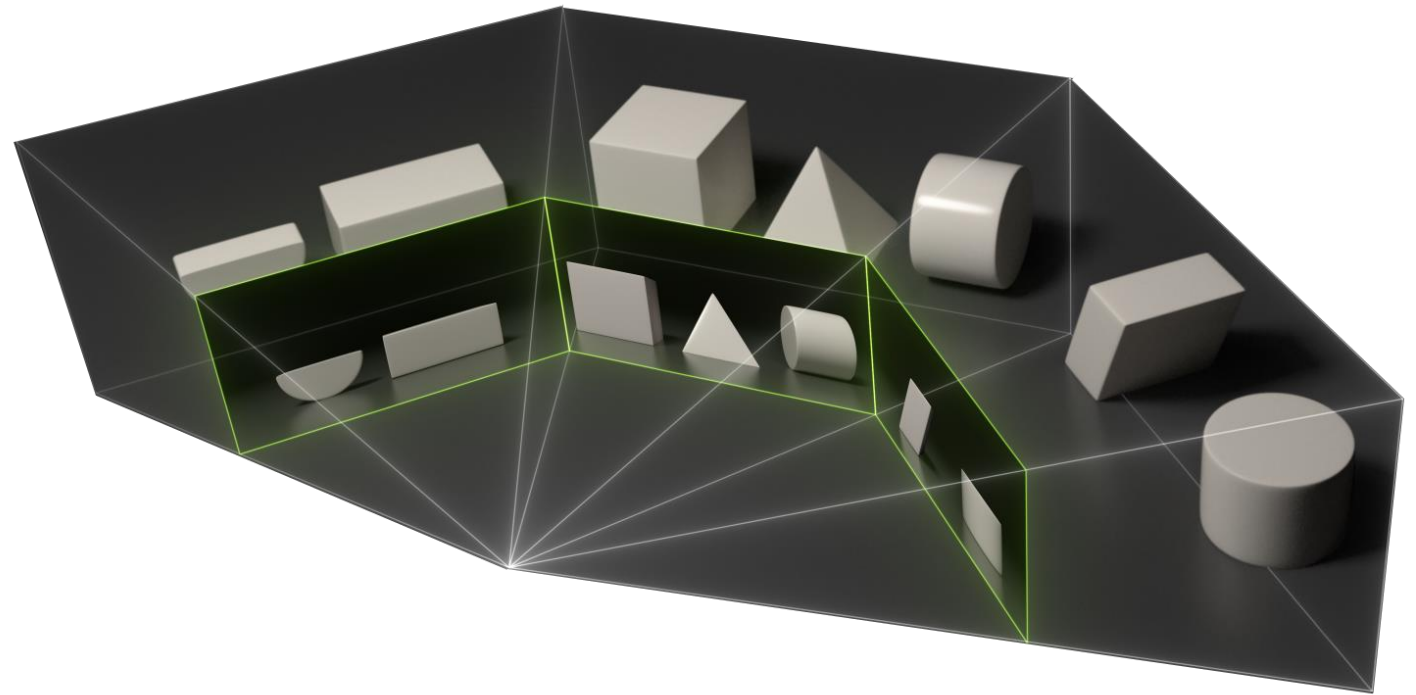
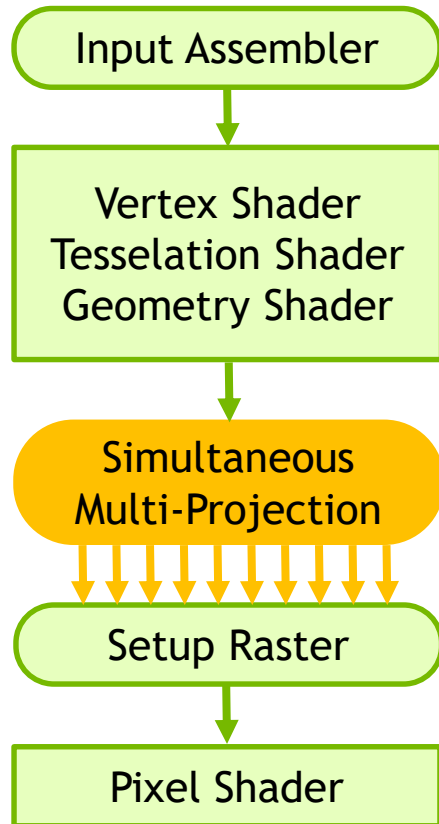


Right Eye (Pass 2)



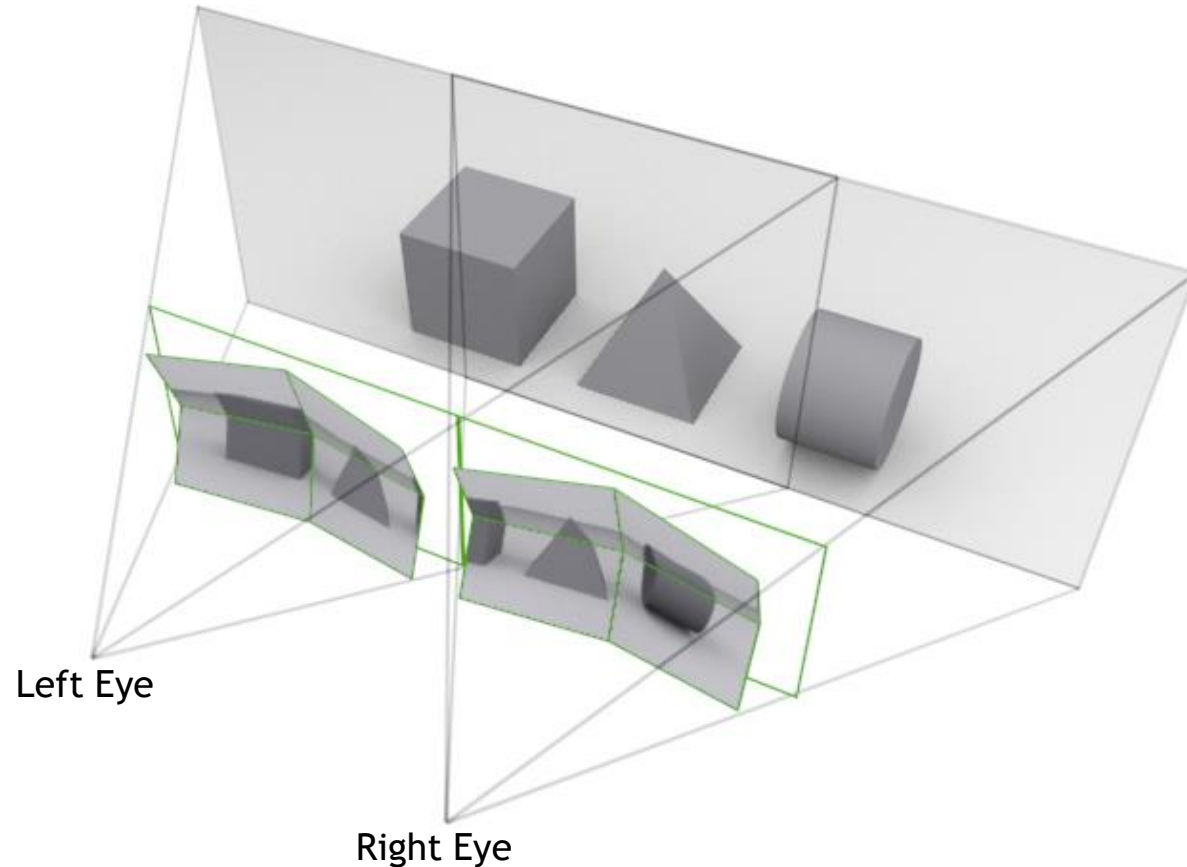
# NVIDIA PASCAL

- Simultaneous Multi-Projection Engine



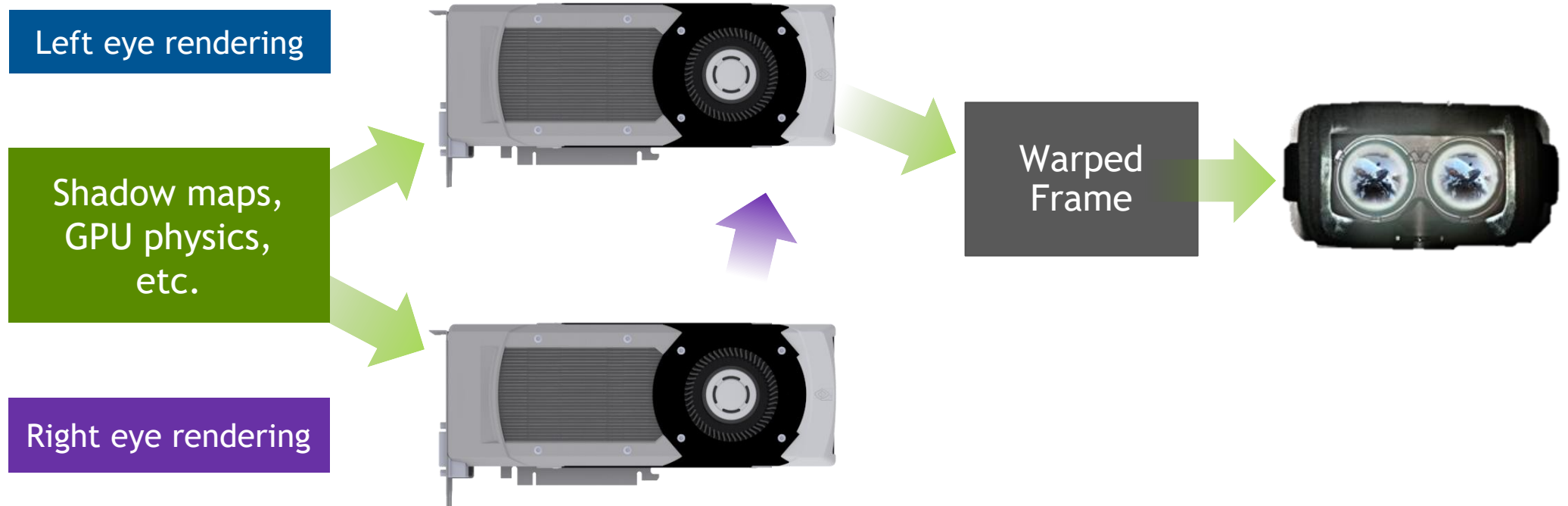
# VRWORKS SINGLE PASS STEREO

Renders left & right eye in one geometry pass



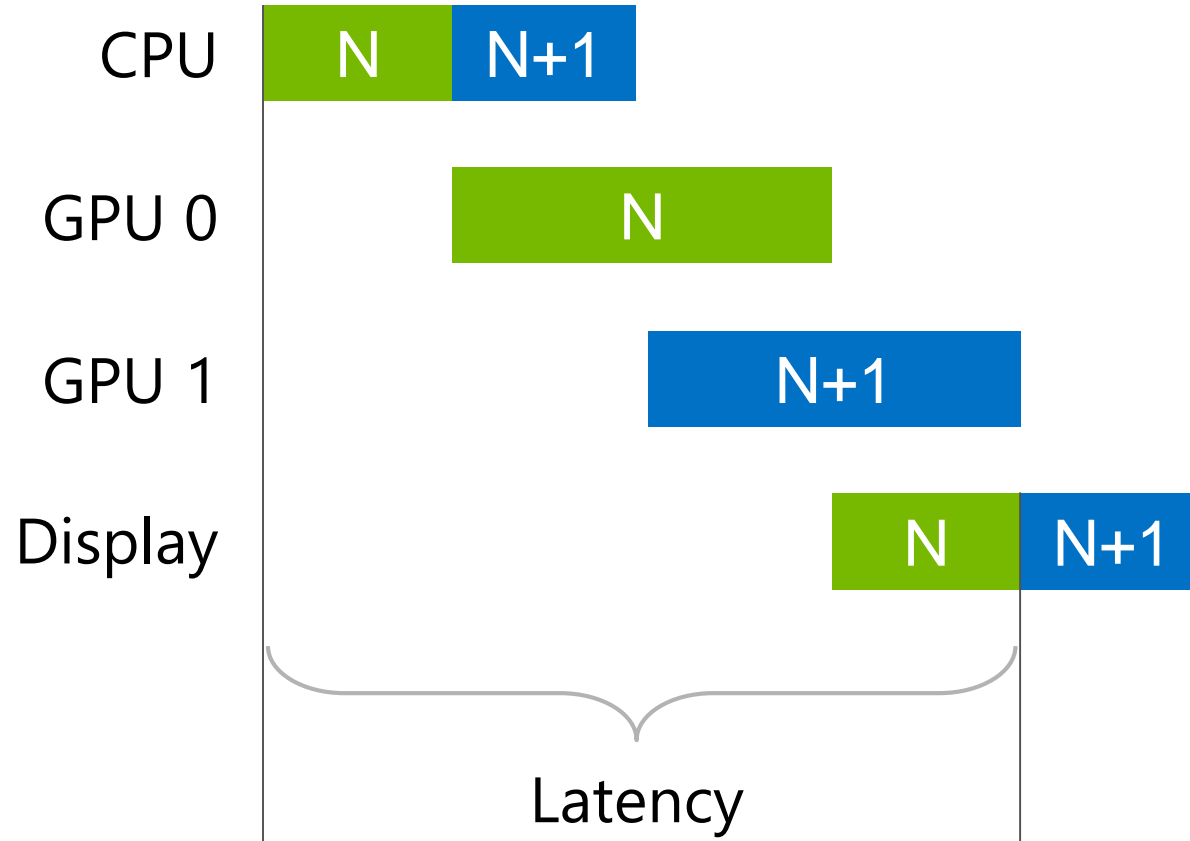
# VRWORKS VR SLI

Scales performance across multiple GPUs



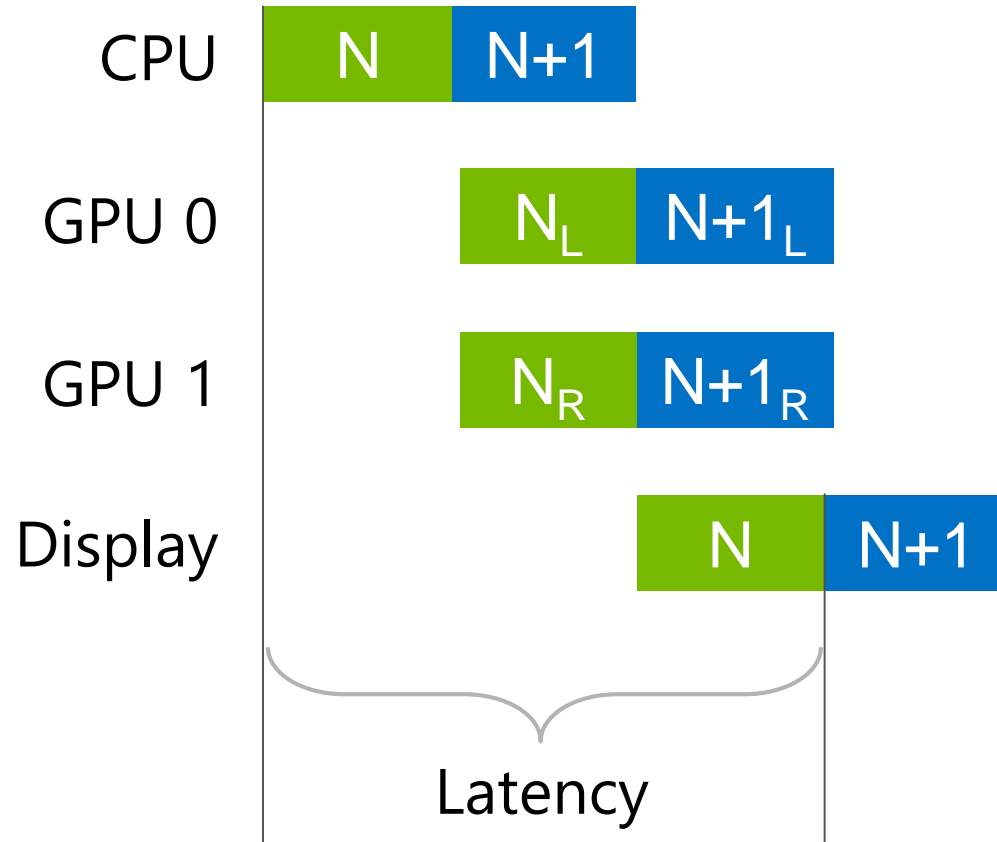
# “Normal” SLI

GPUs render alternate frames

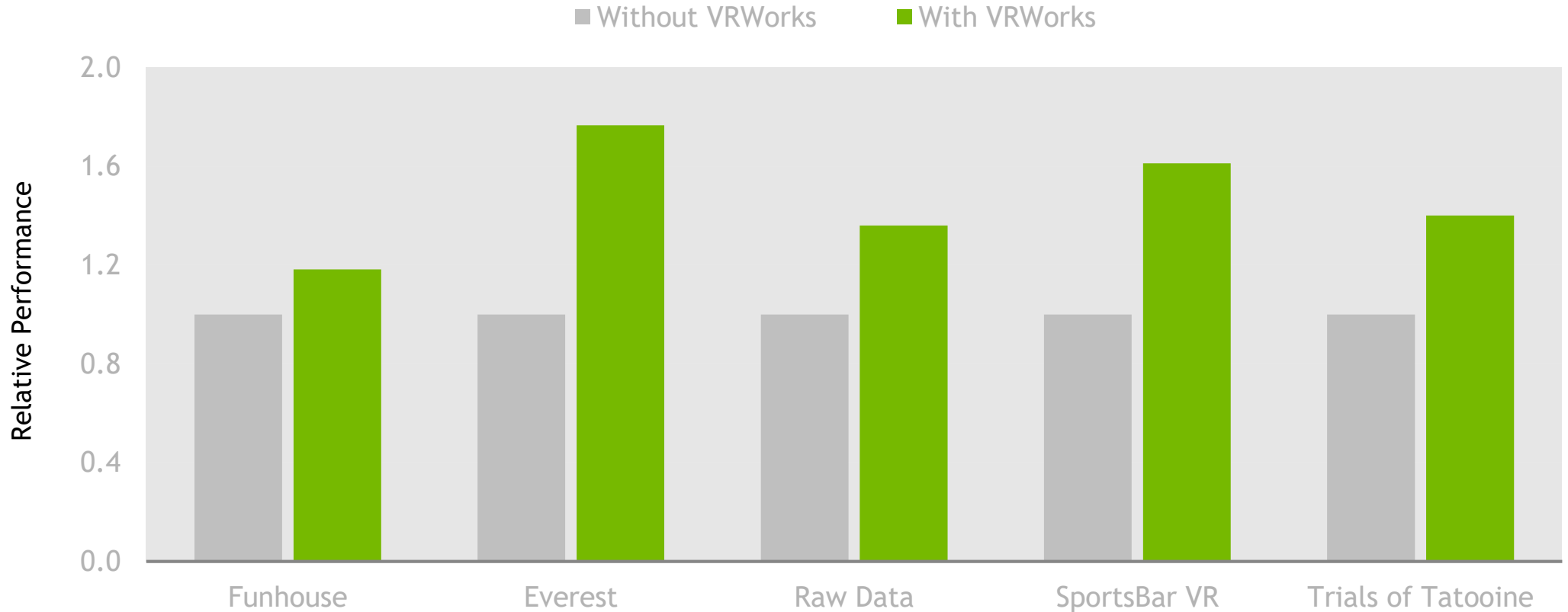


# VR SLI

Each GPU renders one eye—lower latency



# VRWORKS SPEEDUPS





Eco-system

# VRWorks Graphics Support

- Engines

- UnrealEngine 4
- Unity

- APIs

- Direct3D (11 and 12)
- OpenGL
- Vulkan



# VRWorks for Unreal Engine

## Unreal Engine integration

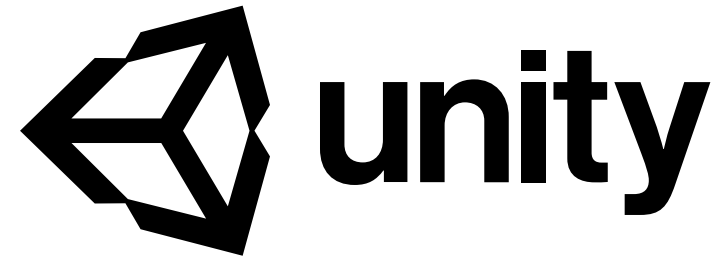
- Full VRWorks suite available
- VRSLI, Multi-resolution Shading, Single Pass Stereo, Lens Matched Shading
  - <https://github.com/NvPhysX/UnrealEngine/tree/VRWorks-Graphics-4.18>
  - Most post passes, instanced stereo supported
- 4.19 coming soon



# VRWorks for Unity

Available in Unity 2017.1 and higher

- Implemented as a native Unity plugin
- Supports MRS, SPS, LMS, and VRSLI
- DX11 only, supports basic post processing, forward rendering
- [developer.nvidia.com/nvidia-vrworks-and-unity](https://developer.nvidia.com/nvidia-vrworks-and-unity)



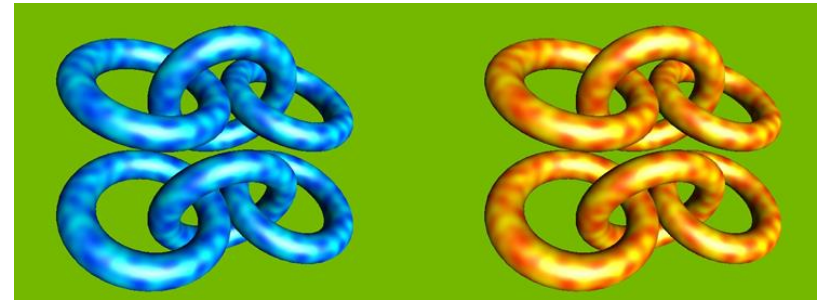
# Vulkan extensions / VRWorks building blocks

- Multi-Resolution Shading (Maxwell+)
  - VK\_NV\_viewport\_array2
  - VK\_NV\_geometry\_shader\_passthrough
- Lens Matched Shading (Pascal+)
  - VK\_NV\_clip\_space\_w\_scaling
- Single Pass Stereo (Pascal+)
  - VK\_NVX\_multiview\_per\_view\_attributes



# Vulkan Multi-GPU for VR

- Vulkan 1.1 / `VK_KHR_device_group_{creation}`
  - Explicit MGPU for AFR, SFR, VR
  - Command buffers & commands can be directed to subsets of devices
  - Viewport/scissor state can diverge between devices
  - Shader built-in `gl_DeviceIndex`
  - Select per eye view transform
- See `vr_sli_vk` sample in VRWorks SDK
- See Jeff Bolz` MGPU talk:
  - <https://youtu.be/RkXa4RiERu8?t=1566>

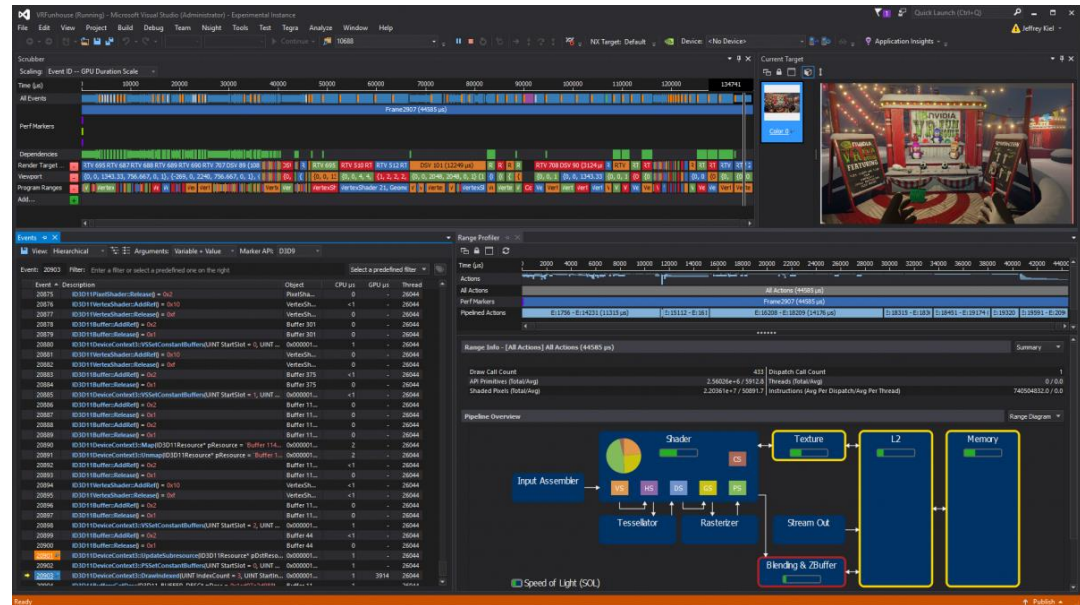


# Measuring Performance

# PERFORMANCE TUNING

## NSIGHT

- Understand CPU/GPU interaction
- Debug your frame as it is rendered
- Profile your frame to understand bottlenecks
- Save your frame for targeted analysis
- Leverage the Microsoft Visual Studio platform
- Also available in the newly released tool, Nsight Graphics



# FCAT VR

MEASURING THE QUALITY OF YOUR VR EXPERIENCE



# PERFORMANCE TUNING

## FCAT

- Create charts and analyze data for:
  - Frametimes
  - Dropped frames
  - Runtime warp dropped frames
  - Asynchronous Space Warp (ASW)
  - Synthesized frames





# NVIDIA VRWorks

Access Latest SDKs at [developer.nvidia.com/vr](https://developer.nvidia.com/vr)

## 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**



**GPUDIRECT  
FOR VIDEO**

# Questions?

[cem@nvidia.com](mailto:cem@nvidia.com)

