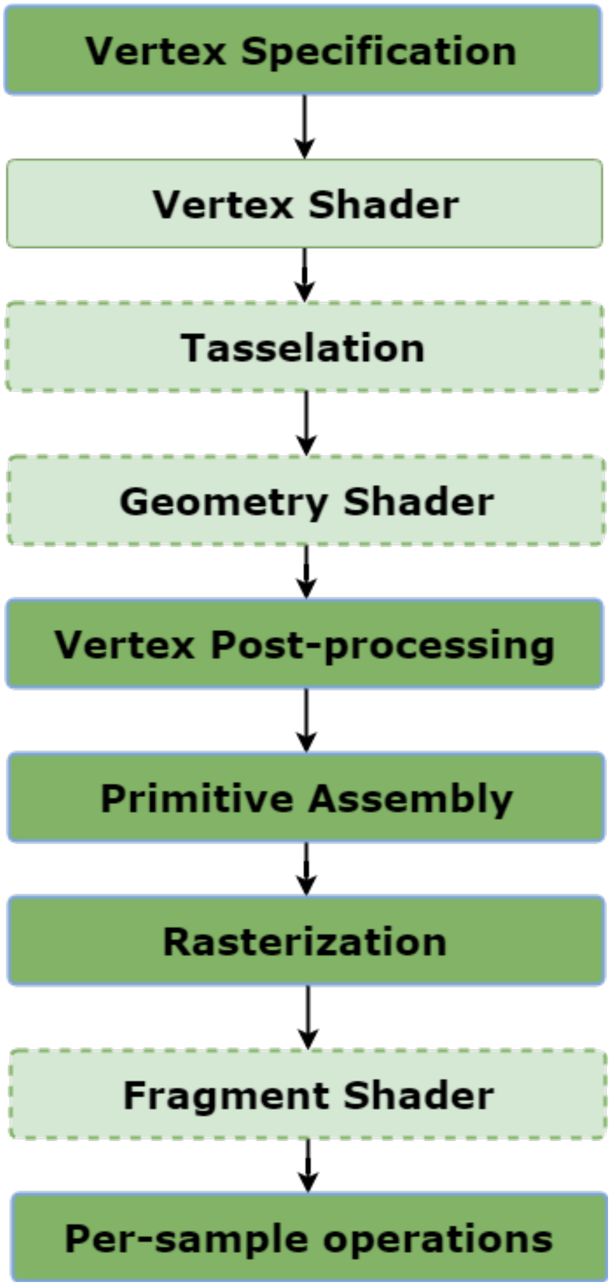
Deferred Shading/AR-VR

How does the Shader Pipeline Work?



Shader Pipeline

Forward Shading Pipeline

- Processes all scene vertices
- Creates all necessary primitives
- Rasterizes primitives to screen based on depth information
- Colors the pixels based on fragment color

Forward Shading Issues

- Considers each object in relation to each scene light
  - Performance issues with increasing light complexity
- Objects processed regardless of whether they are visible to viewer
  - Performance issues with increasing depth complexity

Graphics in the 1990s

Few lights, limited shaders and textures

Modern Graphics

Many lights, complex shaders, many textures

Lighting is Everything

Lights and material properties determine final rendering

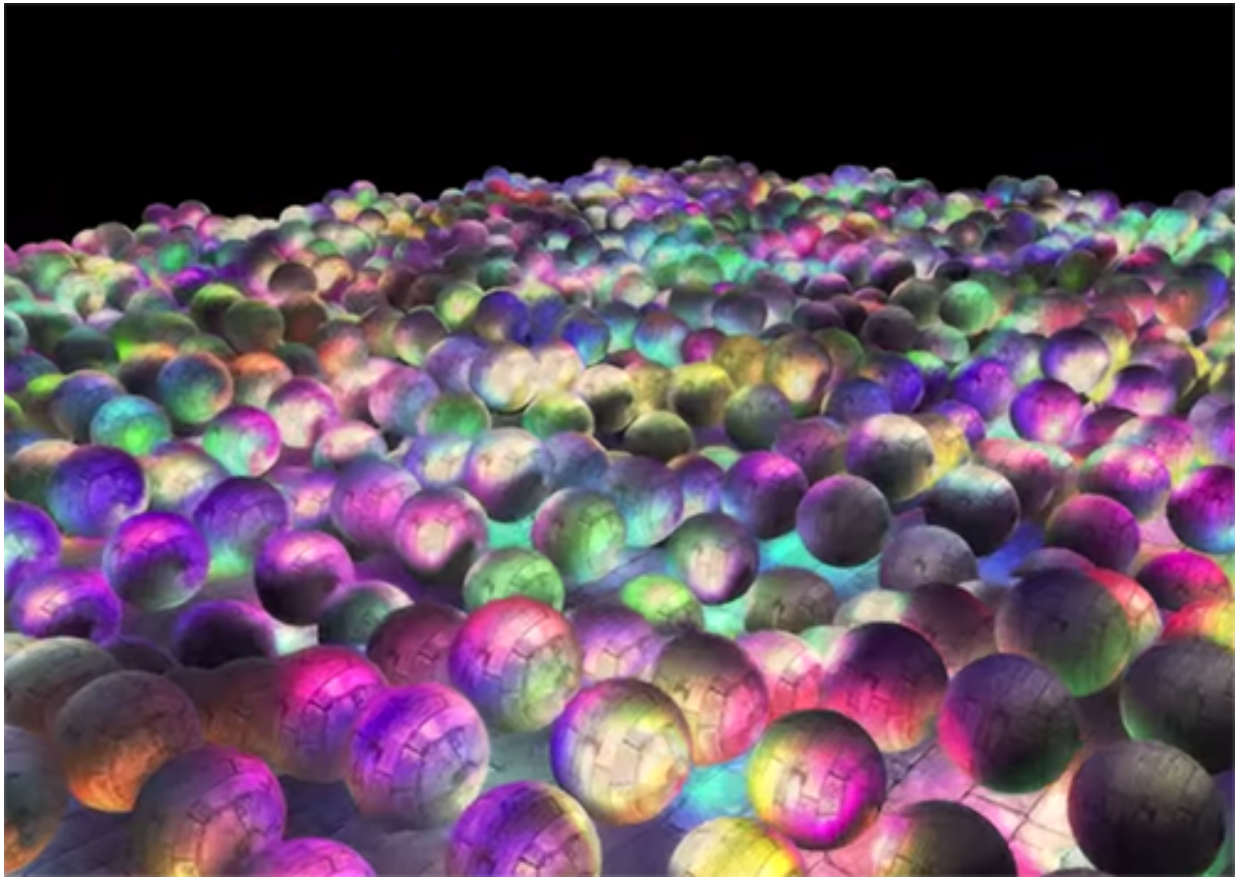
More lights means nicer looking images

How to scale more lights with increasing scene complexity?

Deferred Shading Pipeline

- Defers expensive light calculations till scene complexity is reduced
- Scene geometry considered as textures within the fragment shader
  - Only need to consider scene per-pixel
  - Can better manage light complexity
  - Can be combined with forward rendering and post-processing techniques

Deferred Shading in Action



1847 point lights (Hannes Nevalainen)

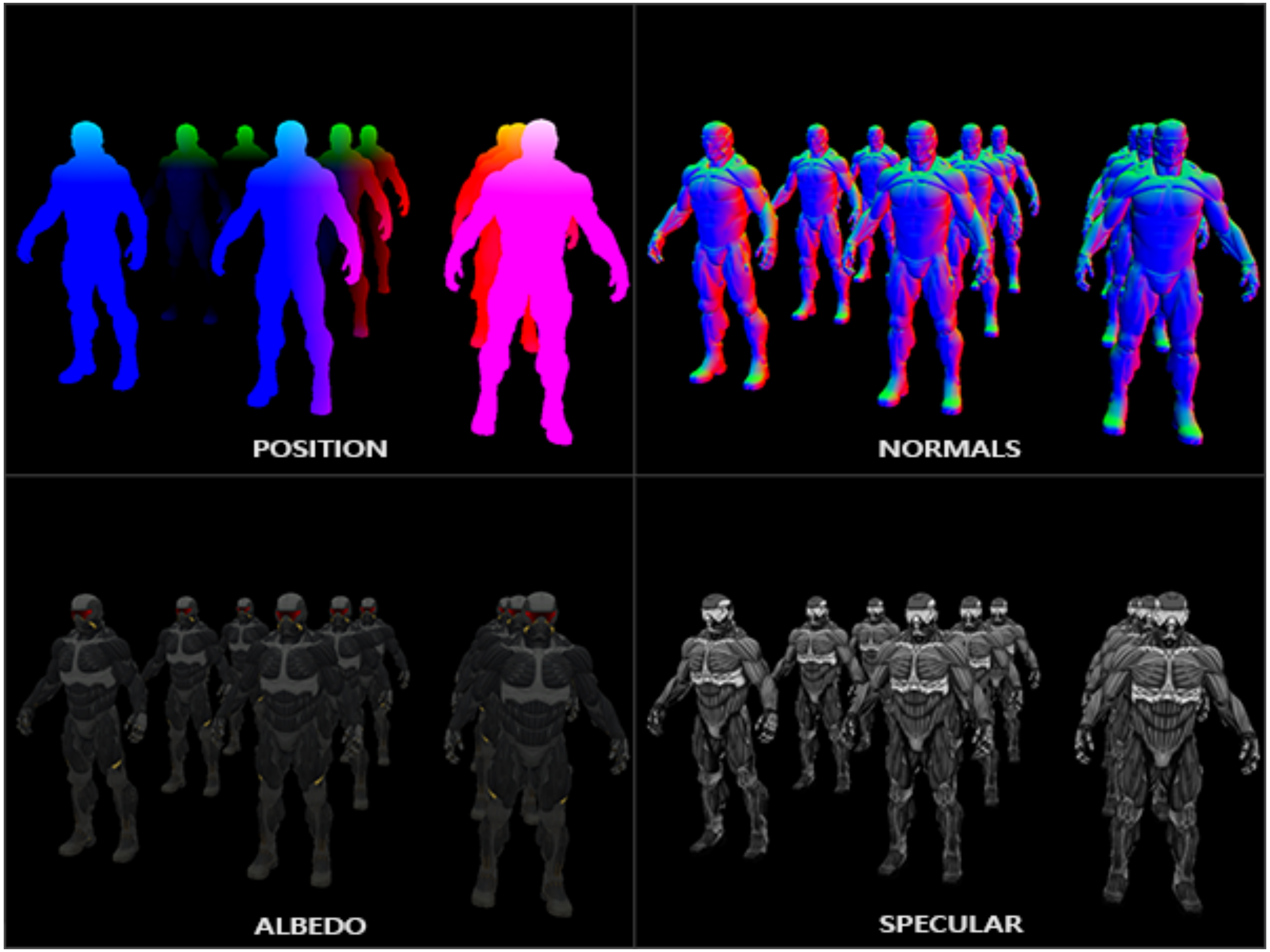
Deferred Shading Passes

- Rasterization broken into two passes:
  - Geometry pass
  - Lighting pass
- Geometry pass stores geometric information in G-buffer
- Lighting pass uses data in G-buffer to reconstruct scene but calculate lights only per-pixel

The G-Buffer

- Contains **textures** holding world-space data needed for final lighting pass
- Depth buffer has already determined which of these is visible per pixel, culling data that is not relevant to scene
- Flexible texture precision allows for compact storage of data

G-Buffer Data



Lighting Pass

- Apply lighting algorithms to G-buffer content rather than scene
  - Only one lighting operation per pixel
- Further optimizations using light volumes
  - Determine a light's "volume" (i.e. how far does it illuminate from the source) based on light attenuation
  - Use of depth and stencil buffer to determine whether light volume intersects front face of given fragment

The Stencil Buffer

- Closely associated with the depth buffer
- Stencil testing allow for the modification or discarding of pixel data based on user-defined conditions
- Expensive but allows for a wide range of screen space effects
  - Shadow volumes, reflections, object silhouettes, etc..

Deferred Lighting

- Adds a lighting pass to the deferred shading pipeline
  - Renders “light shapes” by accumulating diffuse and specular shading
- Can reduce size of G-buffer
- Allows for multi-sample anti-aliasing (MSAA)
- Allows for different shading equations to be applied to different parts of the scene

Translucency and Other Specialized Effects

- Deferred shading cannot handle translucent meshes, as it only considers the closest object per fragment
  - Unable to blend
- A forward pass can be done in conjunction with the deferred pass
  - Takes depth buffer information to determine position of forward pass objects relative to G-buffer positions
  - Allows for blending, special shader effects etc

Deferred Shading and Game Engines

- Most game engines assume a deferred shading pipeline
  - Games want lots of lights and lots of objects
  - Used in UE4, Unity, and most proprietary engines
  - Perform light calculation in multiple stages for non-shadow casting, indirect illumination and shadow-casting lights
- Godot is a notable exception
  - Uses forward shader pipeline
  - Engine designed for mobile development

Combining Raster with Raytracing

- Raytracing is a form of sampling to reconstruct the scene
- **Importance sampling** is a Monte Carlo method for determining the distribution of lighting energy
- Can combine raster pipeline with direct lighting effects using importance sampling

ReSTIR and MegaLights

- ReSTIR is an NVidia technique based on statistical methods that allow reuse of samples in an unbiased way
- MegaLights is an Unreal Engine technique based on importance sampling to handle shadows and light evaluations, replacing deferred shading

UE5 MegaLights demo

Forward Rendering for VR

- Forward rendering is much faster
- Works better with MSAA and does not require full-screen passes
- To allow for more dynamic lighting and scene geometry VR uses:
  - Extensive culling and LODs (level of detail)
  - Emulation of lighting with as few lights as possible
  - Cheap, high coverage directional lights

Why is Fast in VR Important?

VR and AR

- Virtual Reality
  - Creation of a fully immersive environment including stereoscopic vision and haptic feedback
- Augmented Reality
  - Human vision “augmented” with additional information such as visual overlays

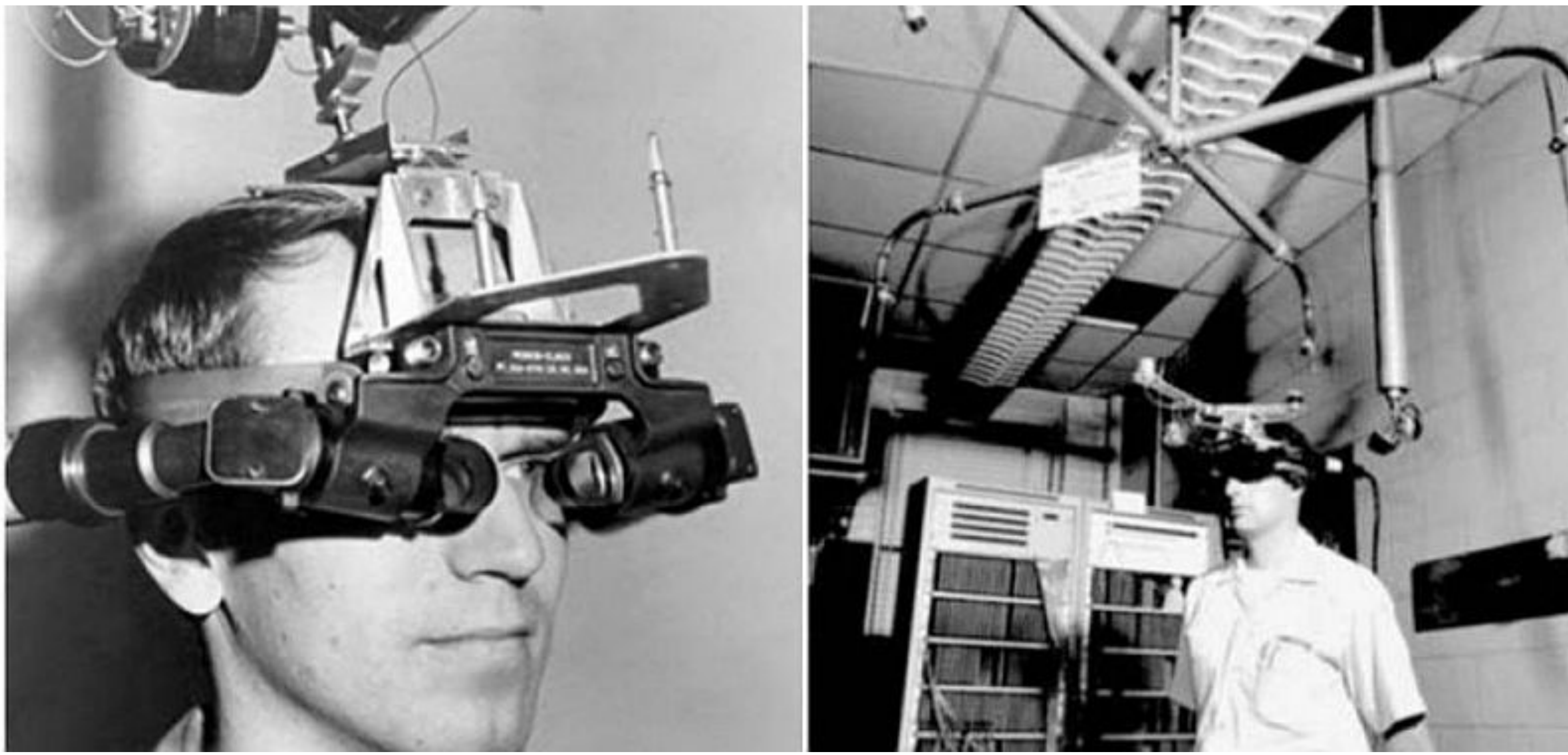
VR and AR Games

Other Applications?

VR Challenges

- Rendering requires low latency and high resolution to prevent sickness
  - Hardware limitations mean software and art solutions required
- Physiological issues related to eye strain, helmet weight, and balance
- Positive interactions require good handling of movement and sensitive haptics

The History of VR



Sword of Damocles (1968)

How VR Works

- Two cameras (close to the eyes) render out scene to create a stereoscopic image
- Fundamentally has not changed since the 60s
- Faster lighter hardware leads to renewed interest in VR at a rate of ~20 years*

* This is a snarky estimate -- not actually confirmed

VR Latency

- Need as low latency as possible to avoid simulator sickness
  - Aiming for 10ms latency
  - Must account for both software and hardware needs (i.e. head-tracking, rendering, display)
- Judder is the smearing/strobing that occurs when the display changes quickly
  - Caused by low refresh rate and high persistence of display
  - Need high refresh rates (120Hz in practice -- ideally 1000Hz) and low persistence (pixel only lit for 2ms)

How to Make Rendering Faster?

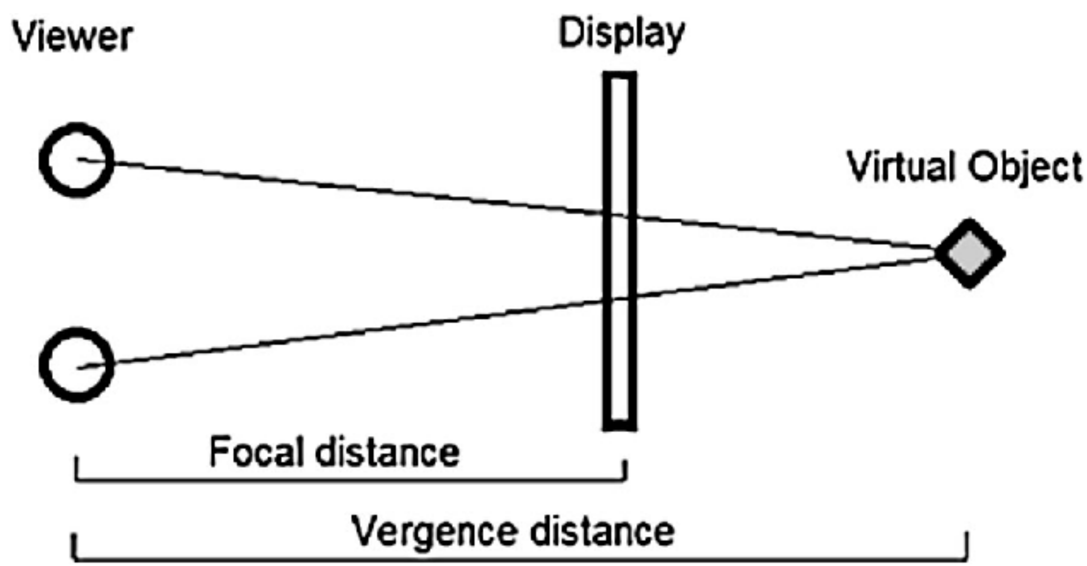
- With good eye-tracking, can better spend rendering budget on **foveated region**
- Humans can only focus on a small region at any given time
- Use of re-projection to take lower frame rate rendering and synthesize new frames at a higher frame rate to match head movement
- Note: many fast-rendering “hacks” such as billboarding and normal mapping don’t work in VR so require more expensive techniques...

VR and Human Vision

- Camera must be close to the viewer's eye for stereoscopic effect to work
- Accommodation is the process where the eye changes optical power to maintain focus at multiple distances
- Vergence is the simultaneous movement of eyes to maintain binocular vision
- Accommodation-vergence reflex allows eyes to automatically adjust focus on objects based on distance

Accommodation-Vergence Conflict

- Brain receives mismatching cues between distance to the object and focal distance of the screen
- Results in conflicting depth cues
  - Blurry image, nausea, fatigue, etc...



Solving for Accommodation-Vergence Conflict

- Most modern solutions focus on changes to hardware
- UX/game designers can also build out content with this idea in mind
- Light fields describe the amount of light at any point in space (holographs)
  - Results in an image that is autostereoscopic and more similar to viewing the actual scene
  - Ideal for VR but requires a lot more camera data (i.e. need better hardware!)

AR Challenges

- Relies heavily on computer vision techniques to understand scene information and correctly project and order augmented data
- Need for image segmentation, image recognition, and depth reconstruction

NVidia Digits 5