CS354P

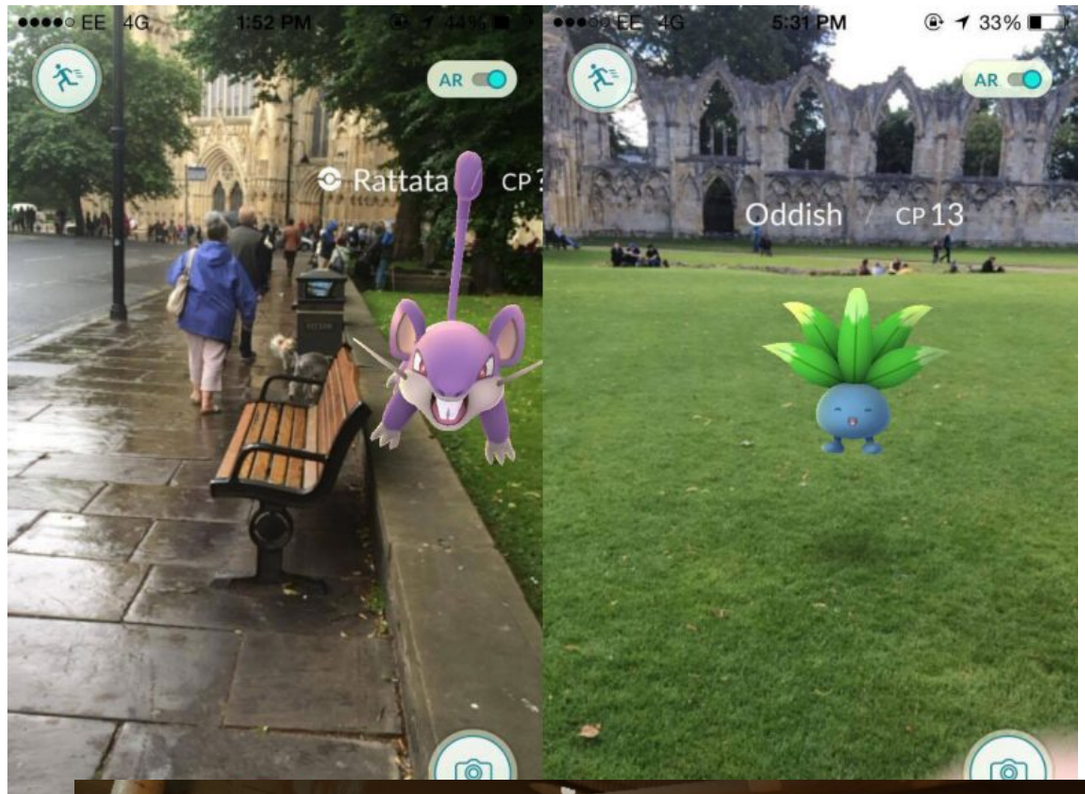
DR SARAH ABRAHAM

AR/VR

WHAT IS AR/VR?

- Augmented Reality
  - Human vision “augmented” with additional information such as visual overlays
    - Practical and game applications?
- Virtual Reality
  - Creation of a fully immersive environment including stereoscopic vision and haptic feedback
    - Practical and game applications?
- We will mostly talk about VR today

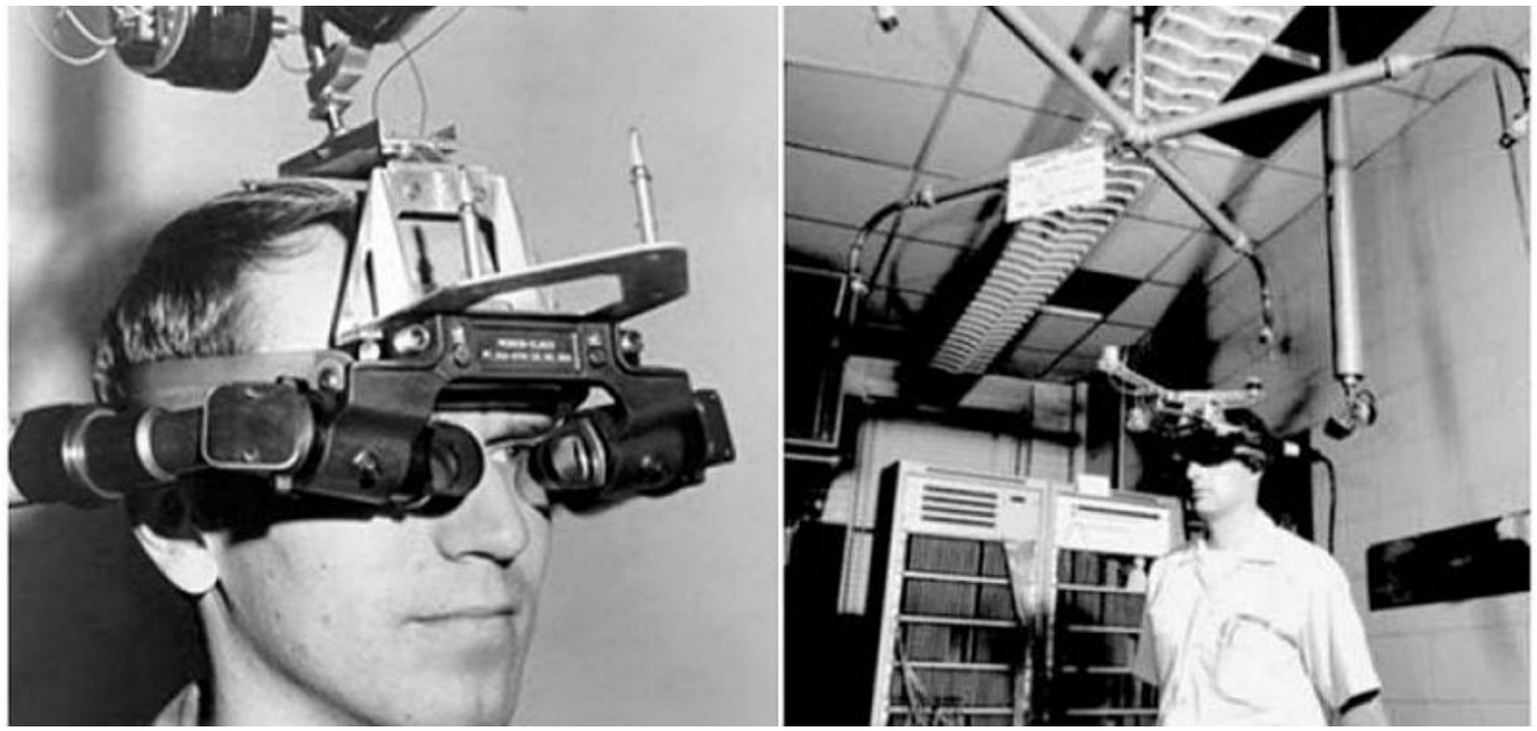
Pokemon Go



Half-Life: Alyx

VR CHALLENGES

- Rendering
  - Need for low latency
  - Need for high resolution
    - Hardware limitations
- Physiological
  - Eye strain
  - Helmet weight
  - Player balance
- World Interaction
  - Movement
  - Haptics



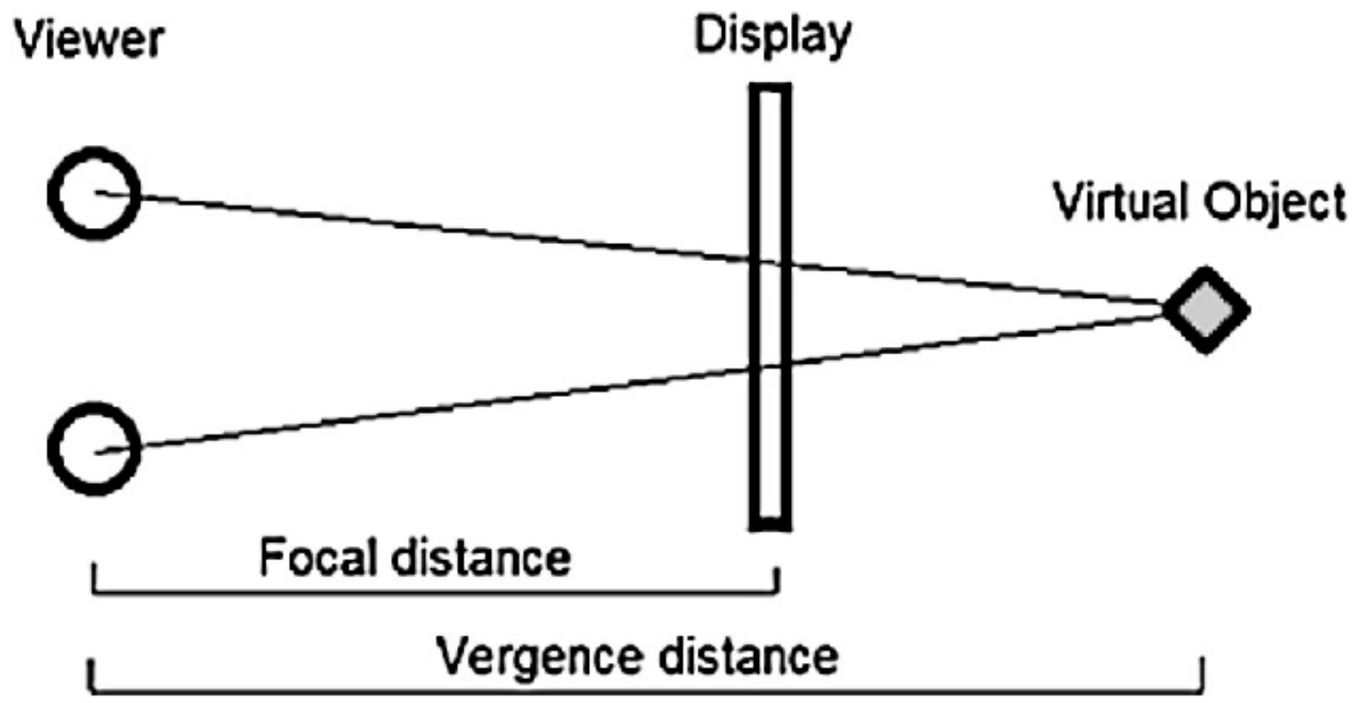
Sword of Damocles (1968)

ACCOMMODATION AND VERGENCE

- 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...



HOW TO RESOLVE THIS?

- Hardware solutions
  - Mechanically adjustable focus using additional relay lenses
  - Deformable mirrors to project low-laser light beam into pupil
  - Liquid crystal lens for adjusting optical power based on focal plane
  - Microlens arrays, parallax barriers, pinlight displays, etc…
- Design choices
  - Pick long distance focal cues
  - Move objects at pace that allows for eye adjustment
  - Map simulated distance to focal distance, create reliable depth cues, etc…

A FUNDAMENTAL ISSUE WITH THE TECHNOLOGY

- Still an open problem in VR/AR space!
- Fundamentally, modern VR has not changed since the 60s
  - Faster, lighter hardware, but same principles of rendering from two cameras to create a stereoscopic image
- **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 still better hardware!)

VR LATENCY

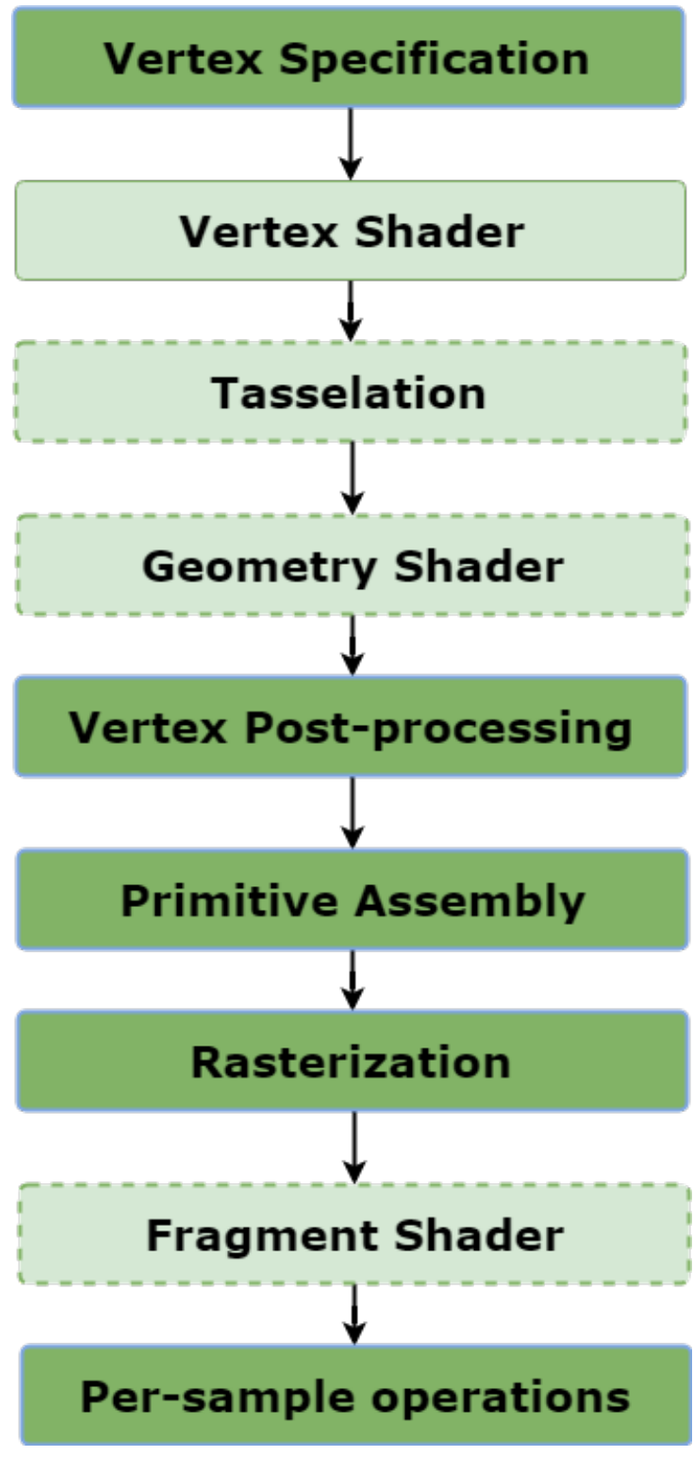
- Need as low latency as possible to avoid **simulator sickness**
  - Aiming for 10ms latency
  - Must account for both software and hardware needs (i.e. head-tracking, rendering, display)
- **Judder** is the smearing/strobing that occurs when the display changes quickly
  - Caused by low refresh rate 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?

- Many fast rendering “hacks” such as billboarding and normal mapping don’t work in VR so already require more expensive techniques
- 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

UNDERSTANDING RENDERING

- Forward shading pipeline:
  - Processes all scene vertices
  - Creates all necessary primitives
  - Rasterizes the primitives to a screen based on depth information
  - Colors the pixels based on the fragment shader



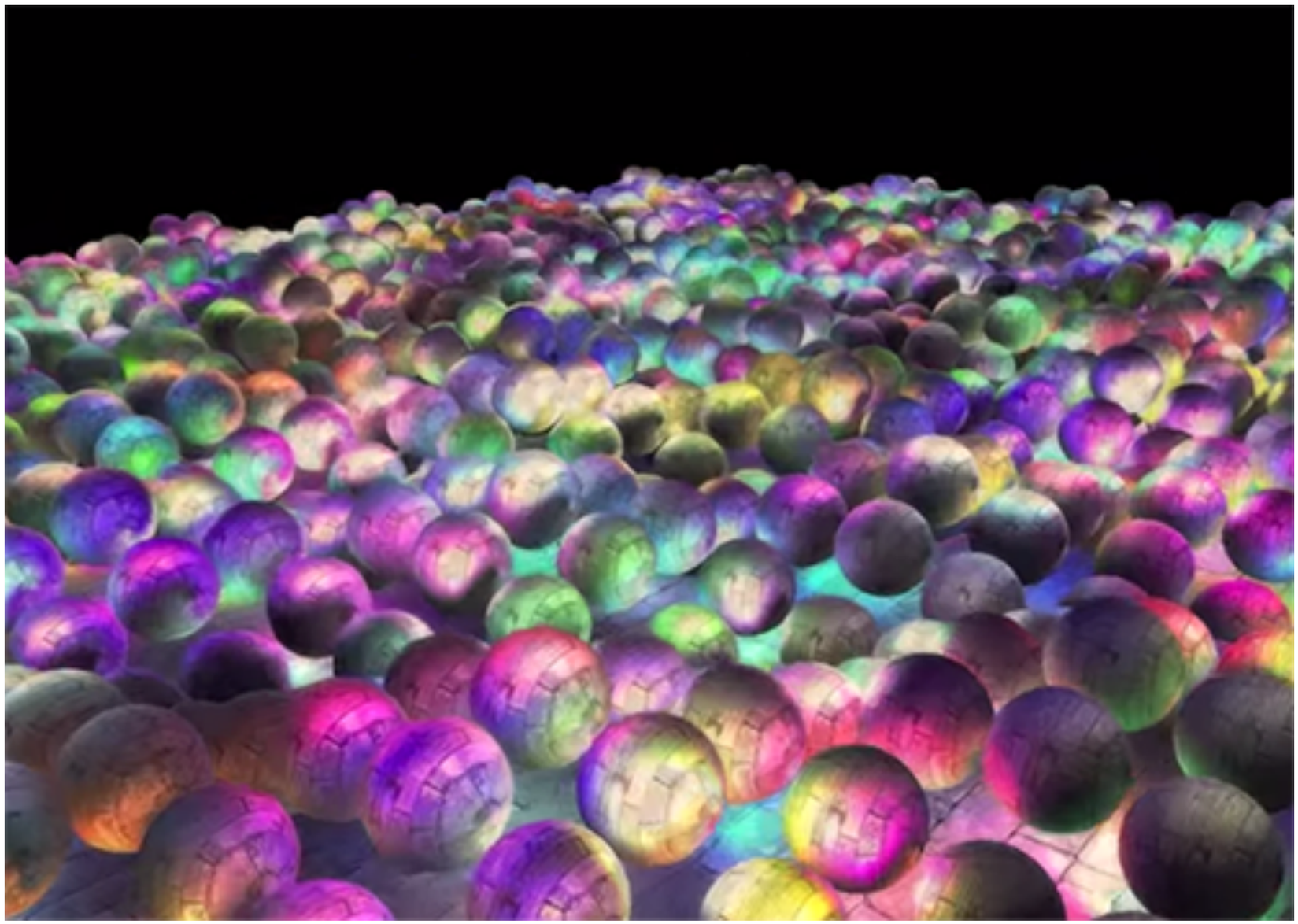
FORWARD SHADING ISSUES

- 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

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



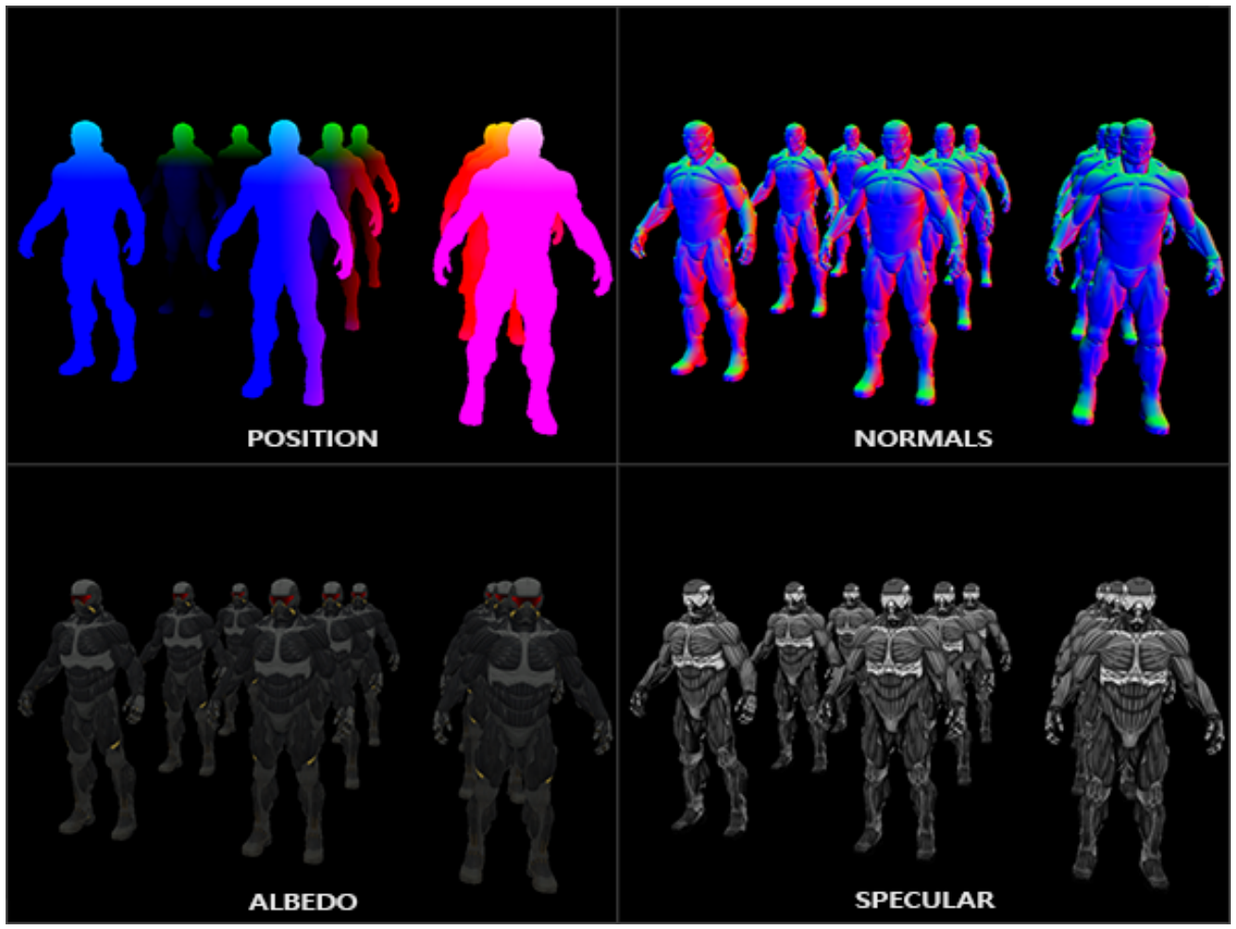
1847 point lights (Hannes Nevalainen)

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



LIGHTING PASS

- Apply lighting algorithms to content in G-buffer rather than to the 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

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

- Game engines generally assume a deferred shading pipeline
- Unreal determines required data for G-buffer based on enabled features and lighting models
  - Standard shader takes values for BaseColor, Metallic, Specular, MaterialAO, and Roughness
  - Modified based on models such as Subsurface Scattering and Decal projection
- Unreal performs light calculation in multiple stages for non-shadow casting, indirect illumination and shadow-casting lights

FORWARD RENDERING FOR VR

- Forward rendering is much faster but cannot handle as much dynamic lighting or scene geometry
  - Works better with MSAA and does not require full-screen passes
- Use of culling and LODs (level of detail) to reduce scene size
- Emulation of lighting with as few lights as possible
  - Directional lights are cheap and provide good coverage

AR DISPLAYS

- Two primary methods of display in AR:
  - Optical see-through
  - Video see-through
- Optical see-through displays allow light to propagate from real-world
  - Use beam splitters to combine with virtual imagery
- Video see-through displays capture video of real world and combine it with virtual imagery before redisplaying it to viewer

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

USING AR AND VR IN UE5

- Unreal uses ARKit and ARCore for iOS and Android respectively
  - Relatively straightforward to create a Unreal project for either but assumes understanding of the 3rd party libraries
- Unreal supports a number of VR platforms (Google, Oculus, Steam, Samsung, Windows Mixed Reality)
  - Basic general information for VR development here: https://docs.unrealengine.com/en-US/Platforms/VR/DevelopVR/index.html

RESOURCES

- https://medium.com/vrinflux-dot-com/vergence-accommodation-conflict-is-a-bitch-here-s-how-to-design-around-it-87dab1a7d9ba
- https://www.cs.umd.edu/sites/default/files/scholarly_papers/Kramidarev.pdf