VRDC @GDC 2018

CocoVR - Spherical Multiprojection

MAGNOPUS

Luke Schloemer
Lead 3D Artist

Xavier Gonzalez
Senior Rendering Engineer

VIRTUAL REALITY DEVELOPERS CONFERENCE | MARCH 19–20, 2018 | EXPO: MARCH 21–23, 2018 #GDC18

UBM

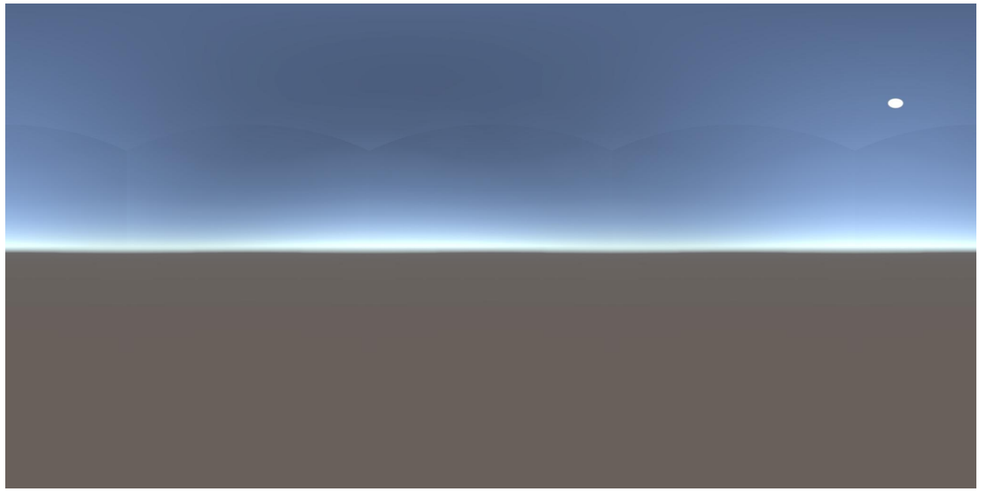
Image(s) courtesy of Disney/Pixar

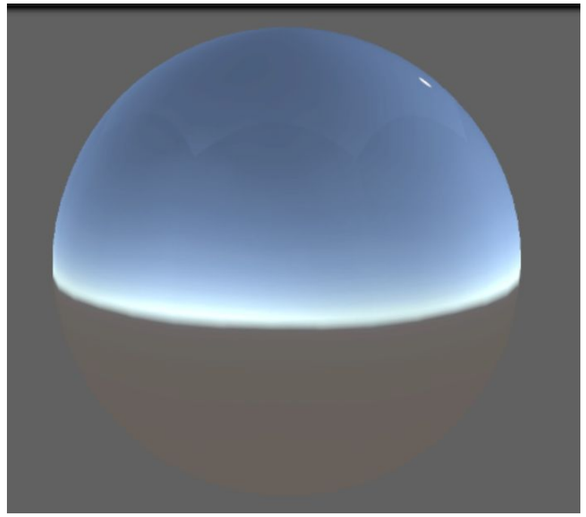
- Production
  - Prototype
    - 3 Months
    - 3-5 Team Members
  - Full development
    - 8 Months
    - 8 Team Members
      - Expanded to 20 near launch.
    - 1 Month in December after launch for 1.1 patch

OUTLINE

1. Intro to Spherical Projections
2. Spherical Projections in Practice
3. Art Pipeline
4. Algorithm Details
5. Tools Developed
6. Conclusion

Intro to Spherical Projections





- Similar to taking a 360 image and applying it to a skydome
- Instead of applying to the background to simulate a sky, we apply it to mostly all geometry in the scene
- A lot of VR experiences are from a single vantage point

mage(s) courtesy of Disney

Spherical Projections in Practice

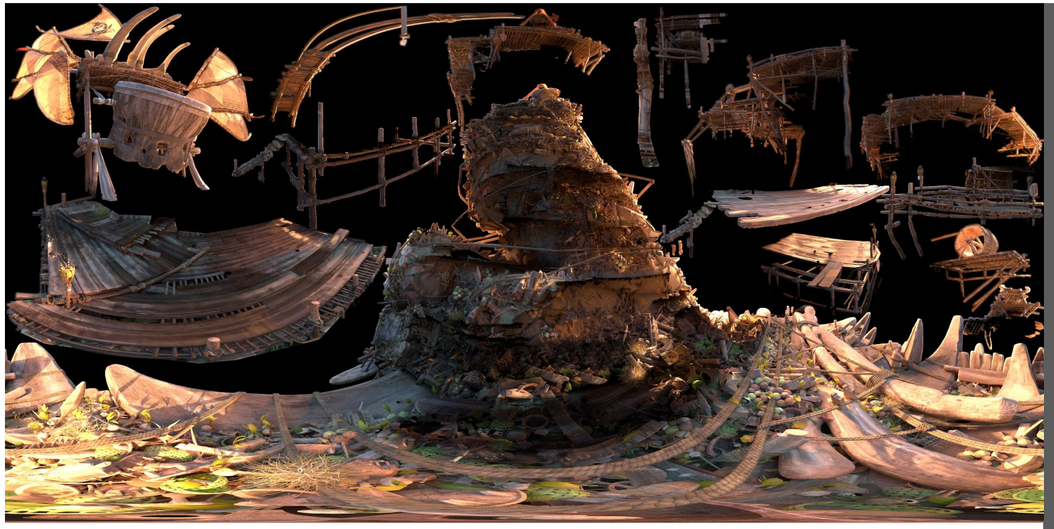
Spherical Projections in Practice

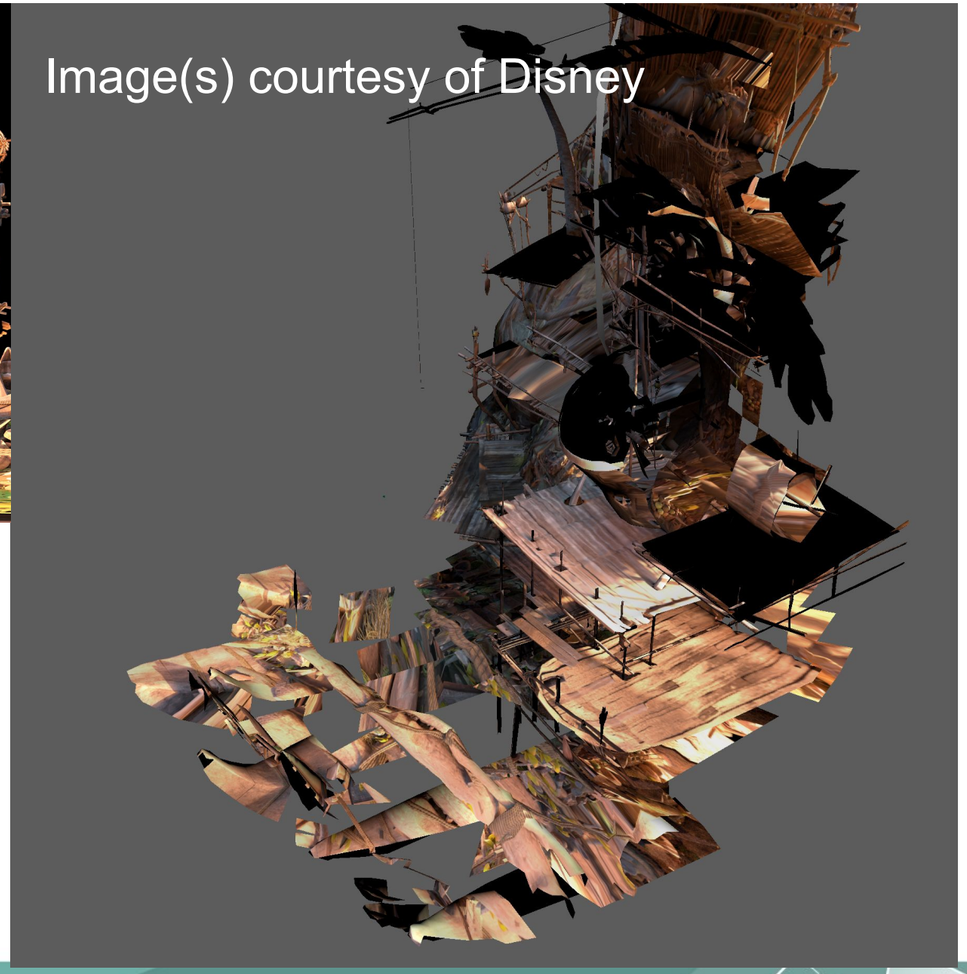
Spherical Projections in Practice

- Moana VR
  - GearVR

Spherical Projections in Practice

Spherical Projections in Practice

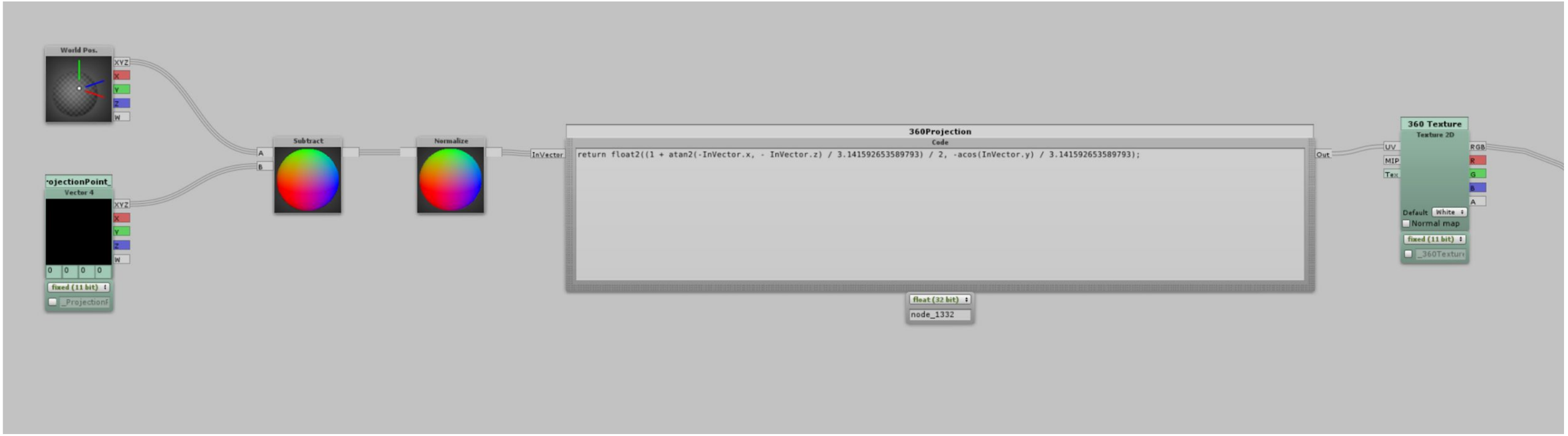




Spherical Projections in Practice

- Coco VR - Prototyping Stage
  - Much more ambitious
  - Walk + Teleport
  - 2 layers, props and architecture

Spherical Projections in Practice



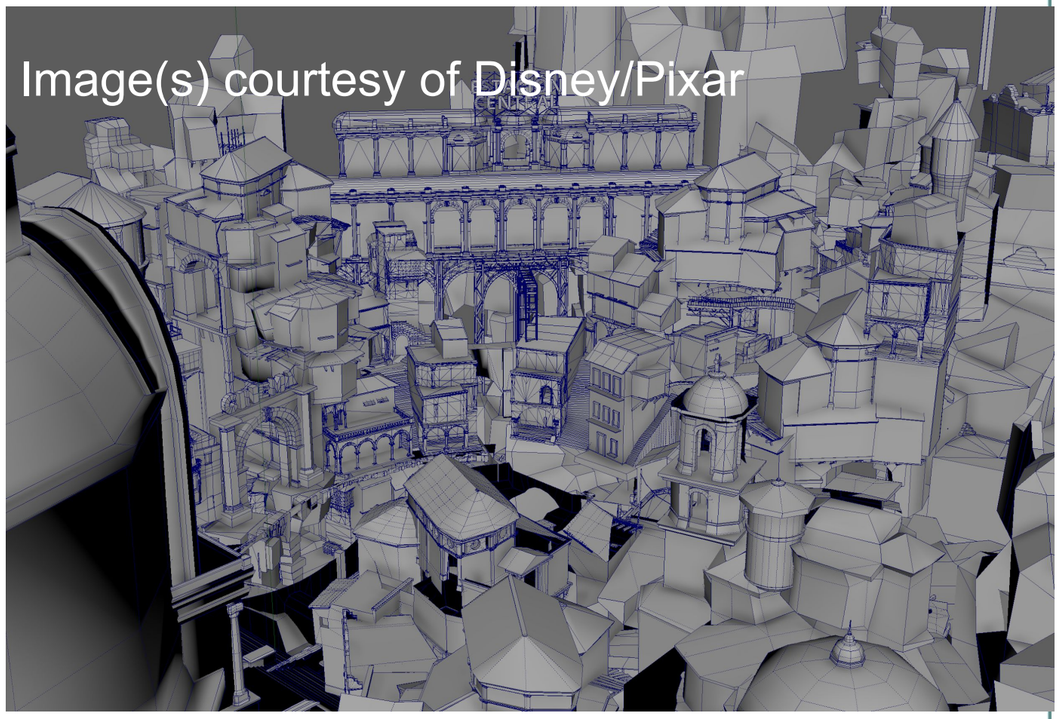
float2((1 + atan2(InVector.x, - InVector.y) / 3.14159265) / 2,
acos(InVector.z) / 3.14159265);

Spherical Projections in Practice

- Coco VR - Prototyping
  - DISCOVERIES!
    - Bypass UV's entirely, do everything on the pixel shader
    - What if? We could blend two projections together?
    - 1st version compared the normal direction w/ location of projection
    - 2nd version used occlusion w/ depth maps to more accurately choose projections

CocoVR - Art Pipeline

- Location Scouting
- We sent Pixar coordinates for the 360's
  - Geometry was final on their end, so we could work in tandem
  - They could update lighting + renders continually.
- Assets with complicated silhouettes were converted into real time assets.
- Entire environments need to be retopo'd by hand (90% of the time)
  - All we care about is silhouette
- Final polycounts can be VERY low



CocoVR - Art Pipeline

- Final Shader does magical things:
  - No separated layers necessary anymore
  - Up to 9 projection points
  - Blends them all together
  - Even supports spec + basic reflections
- Projection Patching

Spherical Multi-projection Shader

Algorithm Overview

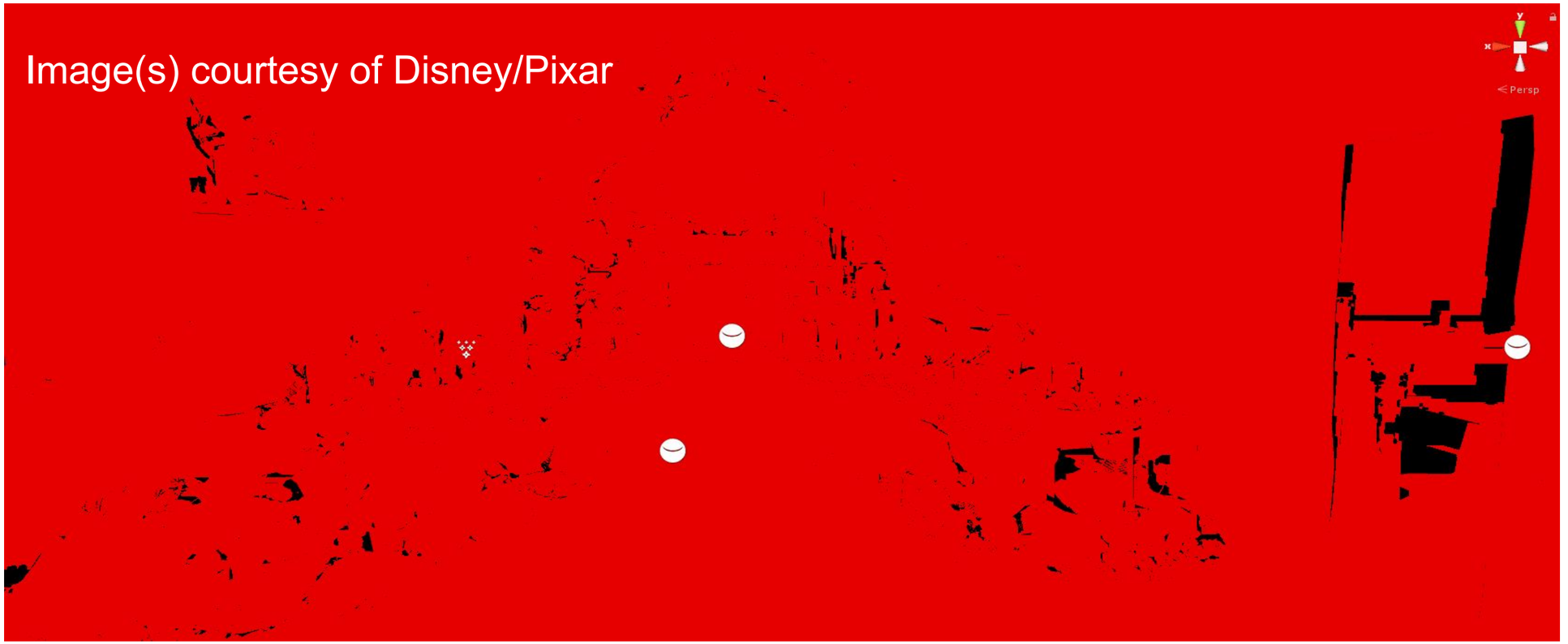
- For each pixel
  - Test visibility against a probe's depth cubemap
    - If probe is visible then run it through a scoring system
  - Project color from the best probe
    - Can also add spec and reflections
  - If no probes were found we can fallback to
    - Flat color
    - Use color from a globally specified probe
    - Use vertex colors to select what probe to use

Visibility Testing

- Every probe contains a depth cubemap rendered from its position
  - Rendered offline
- Unity lacks support for higher precision cubemap formats for storing depth
- Played around with different 32-bit float encoding functions
  - Most I tried introduced too many instabilities in the values
  - Inaccuracies got worse as you moved farther away from a probe
    - Added a small bias to this that increased slightly with distance
- Never got around to testing a latlong texture storing depth values
  - Could potentially solve some or all of the problems with the cubemaps

Probe Visibility Preview

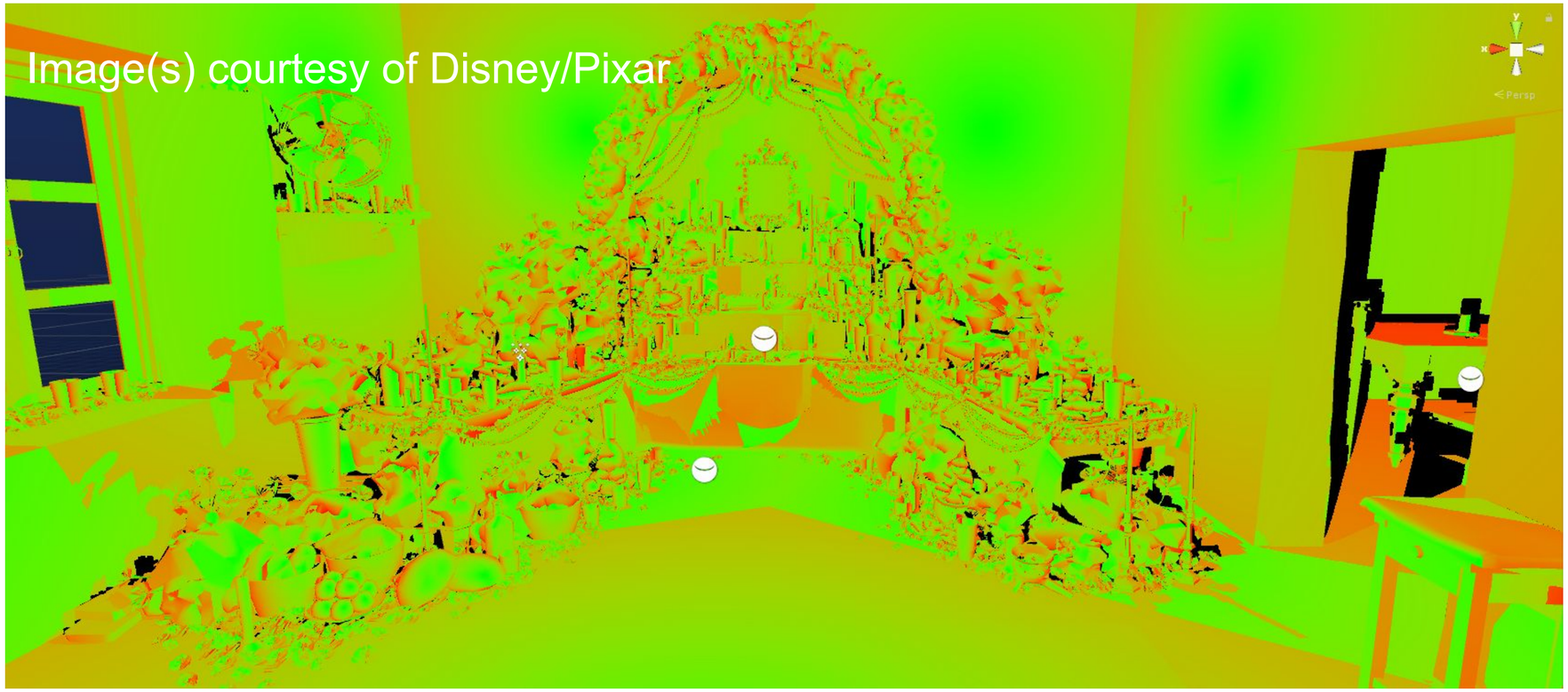
- Probe visibility view mode to help probe placement



Scoring System

- World space distance from probe to vertex
  - Helps reduce blurry texels
- Angle of incidence
  - Helps with projection aliasing
- Each probe contains a weight that determines its contribution to the final image
- Probe contribution can be configured through AABB volumes placed in level
  - More control over the probes used to generate the final image

Angle of Incidence View Mode



Generated UV View Mode

Shaded View

Local Cubemap Reflections

Other Rendering Features

- Cascaded Shadow Maps
- Fog
- Parallax Occlusion Mapping
- GGX specular

When the Search Fails

- Global probe index to use if none are found
  - Can be hit or miss with the results it produces
- Vertex color
  - RGB can hold a flat color
    - Works great for faraway geometry
  - A can hold a index to probe to use

Additional Editor Tools

- Custom material inspector to handle various shader permutations based on the features supported
  - Can greatly simplify the workflow for the artists
- Small tool to handle texture2D array data refresh through right-click context menu
  - Uses meta file to keep track of what textures comprise the array
  - Speeds up iteration time
- Texture2D array viewer since Unity has no way of visualizing the content of such an asset
- Batch generation of depth cubemaps

Conclusion

- Technique looks great
- Cheap!
  - Most expensive was around 0.8 ms to render per eye
  - Meant we could extend the technique with other cool stuff life reflections
- Can be memory intensive
  - Used 8k textures for the quality we wanted
- Small geometry can be problematic as the depth cubemap didn’t have a high enough resolution
  - Typically around 512