Image Processing

Tech Team Presentation

Contents

- Introduction to Post Processing
- Shading Languages
- Ambient Occlusion
- Physically Based Rendering
- Photogrammetry

Post Processing

What is Post Processing?

- Before the image is rendered to the screen it is run through various shaders to add different filters
- Useful because it allows for more efficient rendering techniques for things such as lighting
- Makes final image look pretty and/or more realistic

Various Kinds of Post Processing

- Used in 3D rendering for both film and video games
  - Ambient Occlusion
  - Physically Based Rendering
  - Depth of Field
  - Bloom
  - Cel-Shading
  - Motion Blur

Ambient Occlusion

- Technique used to approximate the effects of environmental lighting
- Most notable use by Industrial Light and Magic for the movie Pearl Harbor in 2001
    - Used on the CG models to make the CG more realistic in the scene
- Multiple methods with varying implementations and processing expense
- Useful in some situations because it does not depend on light direction so it can be pre computed for static objects

Physically Based Rendering

- Notable usage in Pixar's Monsters University in 2013
- Method of making objects look more natural through light bouncing off of objects
- Metals, plastics, leathers, etc all look more photorealistic because of more accurate light and shadow
- Useful by decreasing time spent creating different maps for a surface depending on the situation
    - Instead of diffuse/specular maps you create a map and give it certain properties that you can apply on an object

Cel-Shading

- First use of cel-shading in video games was Jet Set Radio for the Dreamcast in 2000
- Used to mimic a more cartoonish appearance/artstyle
- Typically more flat shading with less gradients

Other Effects

- Depth of field is used to blur things that are out of focus
- A reproduction of a camera focusing effect
- Useful to provide focus to things even in 3D rendering
- Bloom is used to produce fringes of light beyond bright areas
- Another reproduction of a camera focusing effect
- Very popular effect used in many different games

Post Processing in the Unreal Engine

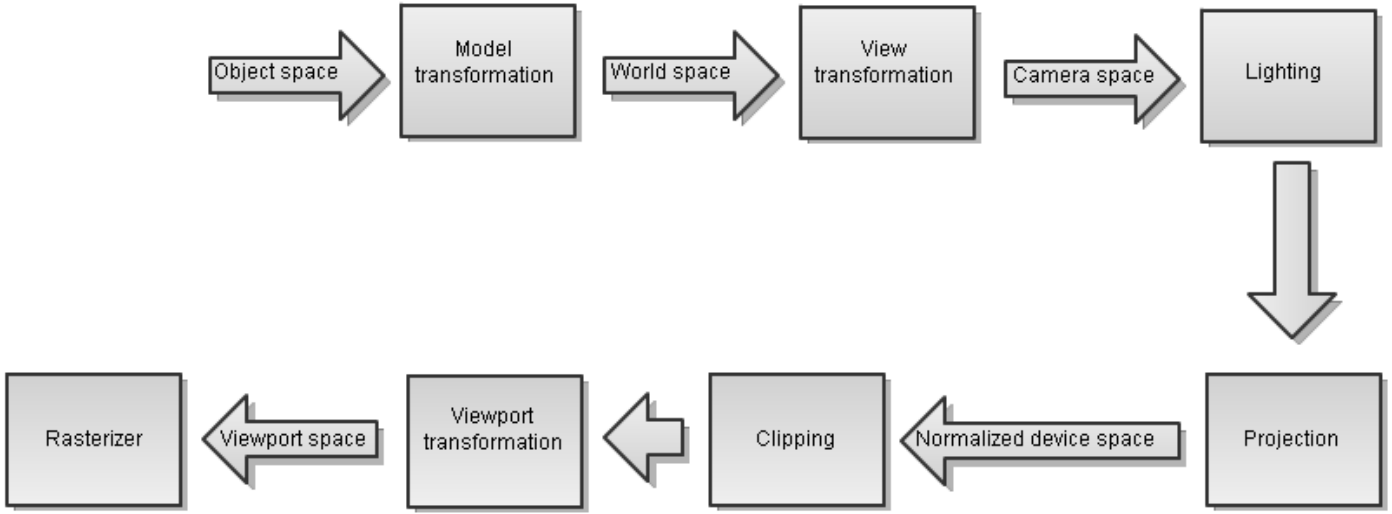
- https://www.youtube.com/watch?v=zzRsPFzu_DY (1 min)

Shading Languages

HLSL and GLSL

- HLSL – High Level Shading Language
- GLSL – OpenGL Shading Language
- HLSL is a proprietary language used for Direct3D
- GLSL is cross platform used for OpenGL
- Both give control over the graphics pipeline and allow PP effects
- Graphics features are mostly similar
- Some differences include
  - GLSL – Fragment Shader, HLSL – Pixel Shader
  - GLSL – Procedural, Similar to C, HLSL – Object Oriented, Similar to C++
  - Performance differences depending on the GPU/drivers

The Graphics Pipeline



Basic vertex/fragment shader code

- Vertex Shader
- uniform Transformation {

```
        mat4 projection_matrix;
        mat4 modelview_matrix;
};
in vec3 vertex;


void main(void) {
        gl_Position = projection_matrix * modelview_matrix *
vec4(vertex,1.0) ; }
```

- Given a vertex vector, applies a certain transformation from local coordinates to 3d coordinates to a 2d camera view on the screen

- Fragment Shader
- void main(void) {

```
        gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0); }
```

- Changes the fragment color to red

Example Unity Shader

- Camera Script

```
// Called by the camera to apply the image
    effect
void OnRenderImage (RenderTexture source,
    RenderTexture destination){

 //mat is the material containing your shader
 Graphics.Blit(source,destination,mat);
}
```

- Partial Frag Shader

```
fixed4 frag (v2f i) : COLOR{
  fixed4 orgCol = tex2D(_MainTex, i.uv); //Get
the orginal rendered color

  //Make changes on the color
  float avg = (orgCol.r + orgCol.g + orgCol.b)/3f;
  fixed4 col = fixed4(avg, avg, avg, 1);

  return col;
 }
```

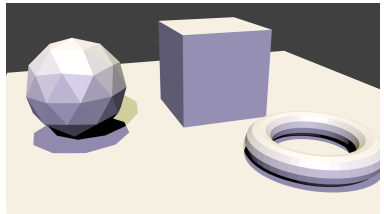
Ambient Occlusion

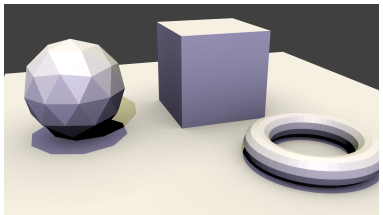
Ambient Occlusion

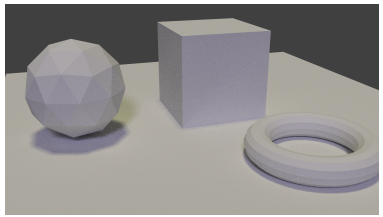
- “True” Ambient Occlusion
- Screen Space Ambient Occlusion (SSAO)
- Distance Field Ambient Occlusion (DFAO)

“True” Ambient Occlusion

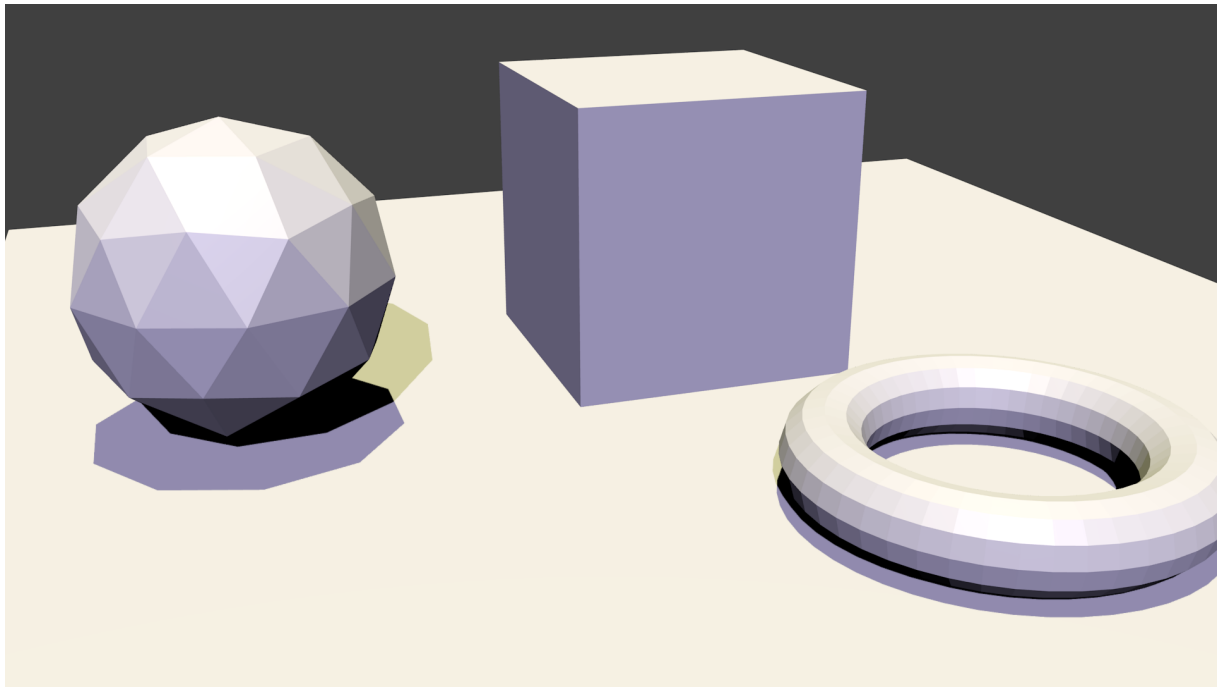
- Approximates global illumination
- Checks distance to surrounding geometry to determine amount of light



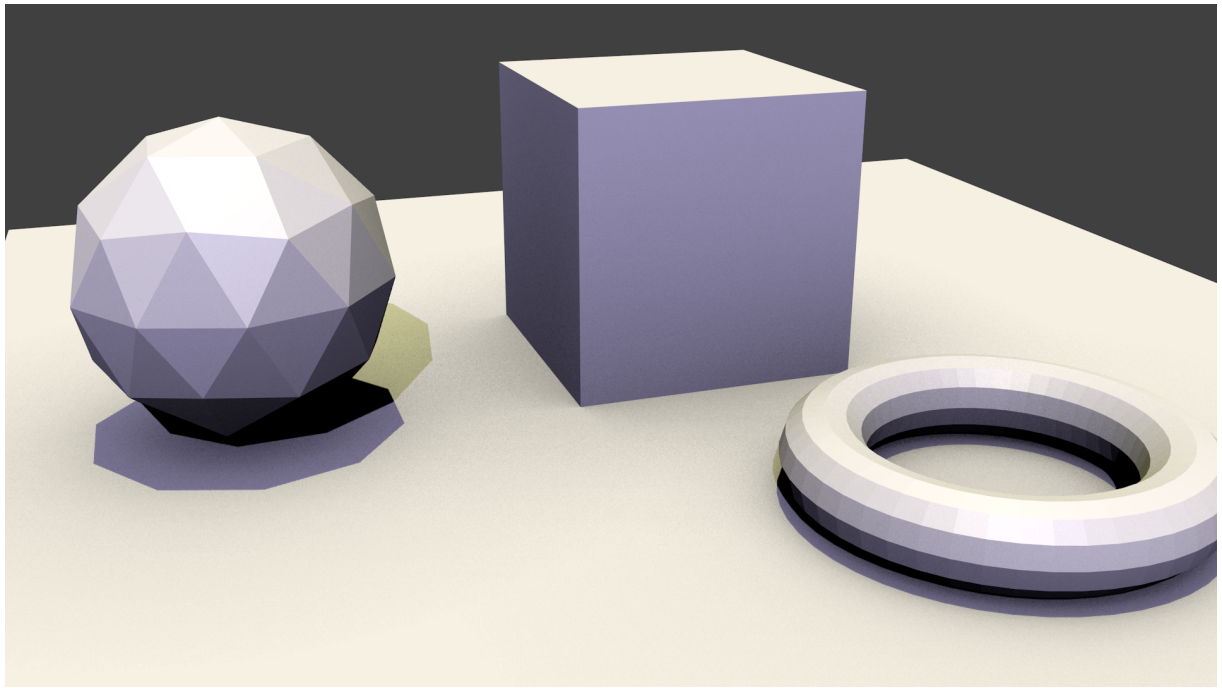




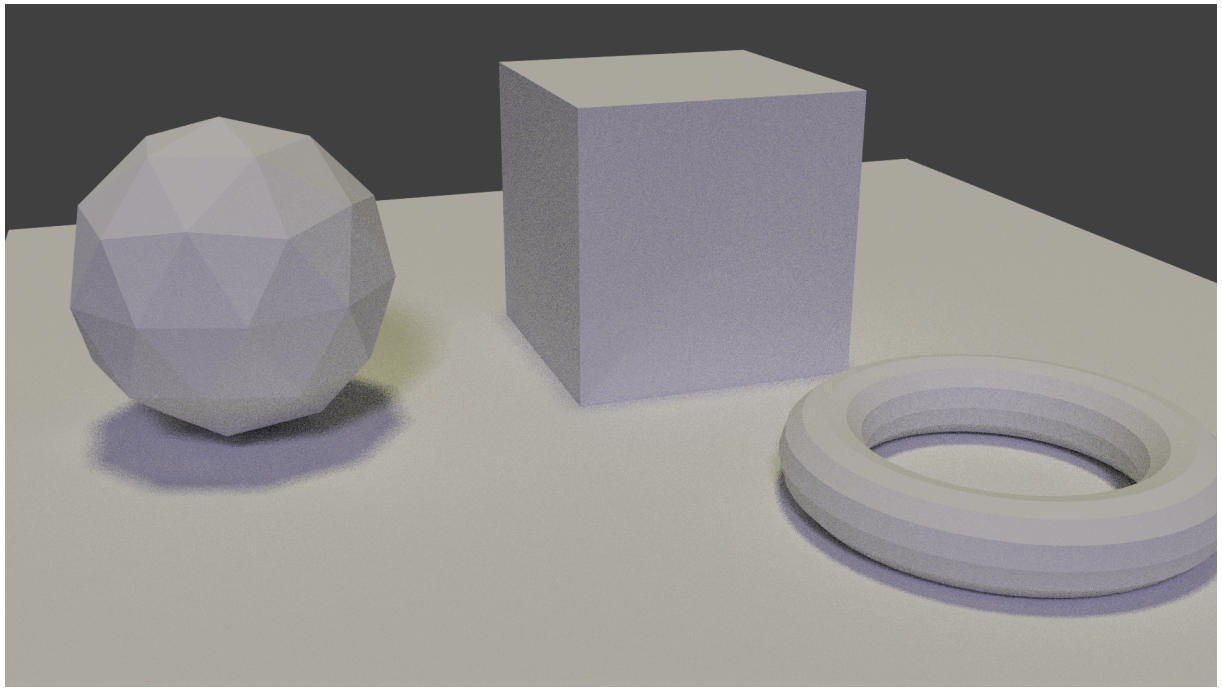
No Ambient Occlusion



With Ambient Occlusion



Global Illumination



Screen Space AO

- Most commonly used form used in games
- Uses the depth buffer to determine occlusion bringing its efficiency to levels comparable to bloom
- Made popular by Crysis in 2007
- Currently supported by both Unity 5 and Unreal Engine 4

Screen Space AO

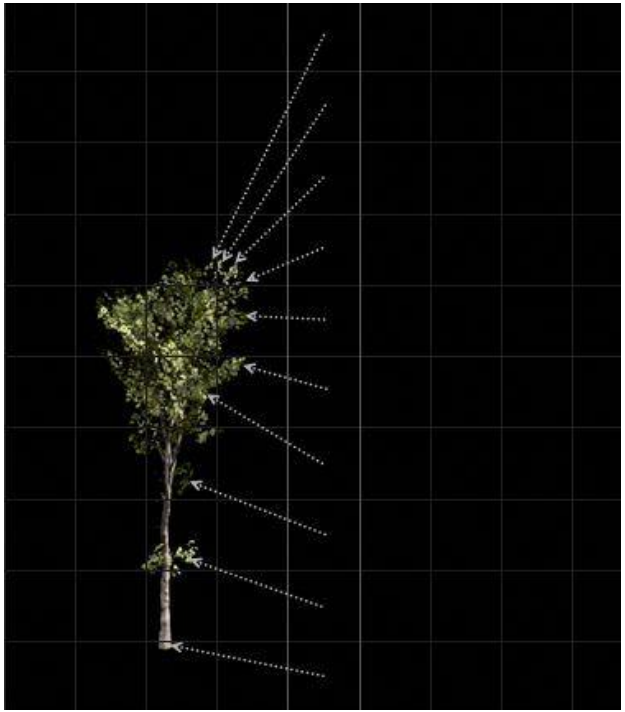
Distance Field AO

- Introduced at GDC 2015 by Epic for Unreal
- More closely approximates “true” ambient occlusion than SSAO
- About 3-6 times as expensive as SSAO
- Works on foliage billboards
- Only works on static meshes (but they can move)

Distance Field AO

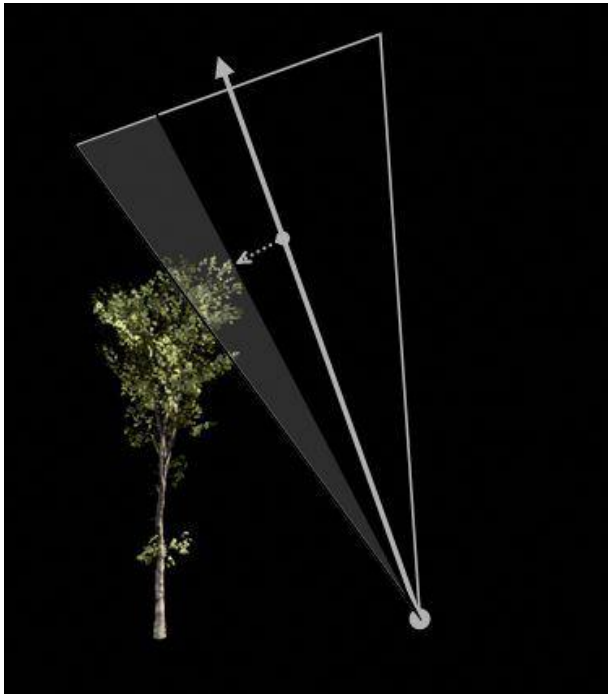
Distance Field AO

- Named “Distance Field” after its use of a vector field storing the distances to the nearest surfaces



Distance Field AO

- Allows for quick computation of conic intersections
- Only a few conic intersections are needed to compute occlusion



Physically Based Rendering

What is it?

- Simulates the realistic interaction between materials and light
- Not photorealism but close
- Everything is shiny
- Everything has fresnel
- (Microfacet) Bidirectional Reflectance Distribution Function (BRDF), Physically Based Shading (PBS)

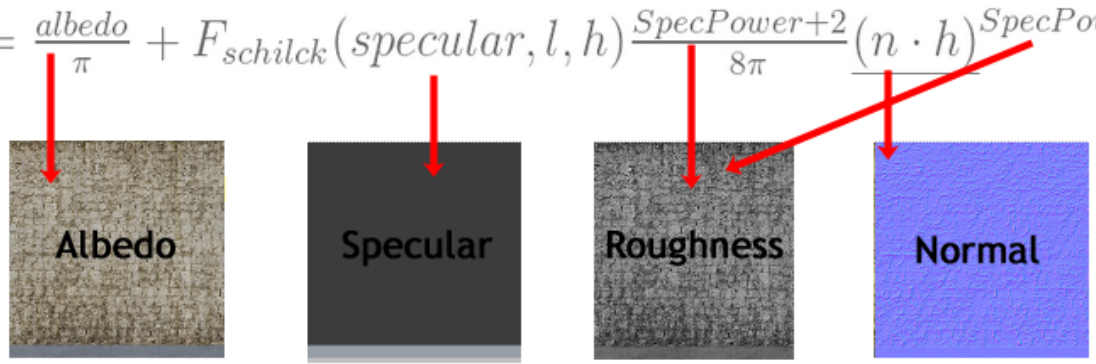
Motivation

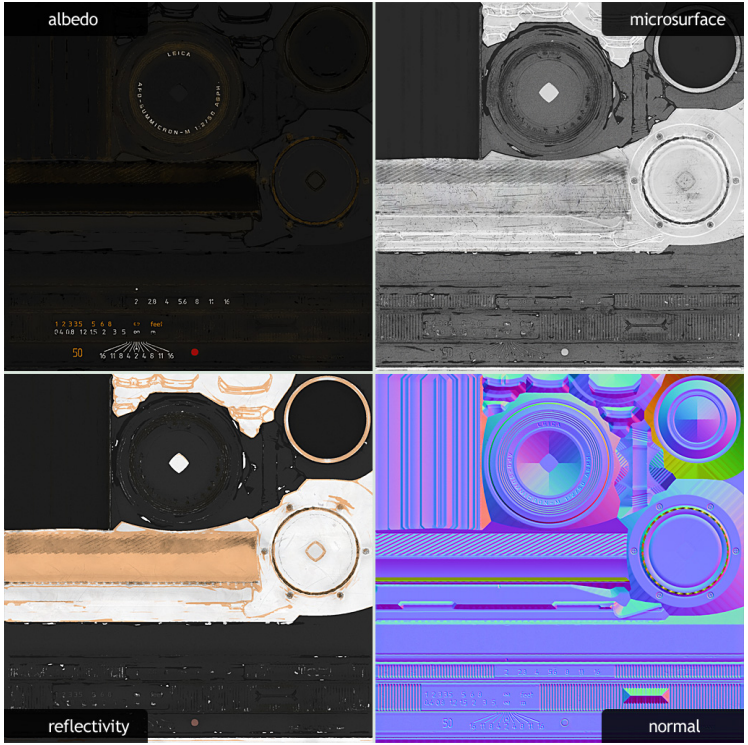
- Unity 4
  - 5 Shader Families
    - Normal, Transparent, Transparent Cutout, Self-Illuminated, Reflective
  - 80 different built-in shaders
    - Vertex-Lit, Diffuse, Specular, Bumped Diffuse, Bumped Specular
- Old system: different lighting = different shading
  - organic, artificial
    - Directional, Area
- More work for artists
- Bake everything

PBR Workflow

- Diffuse/Albedo
- Specular/Reflectivity
- Roughness
- Normal

$$f = \frac{albedo}{\pi} + F_{schilck}(specular, l, h)\frac{SpecPower+2}{8\pi}(n \cdot h)^{SpecPower}$$




albedo
microsurface
reflectivity
normal

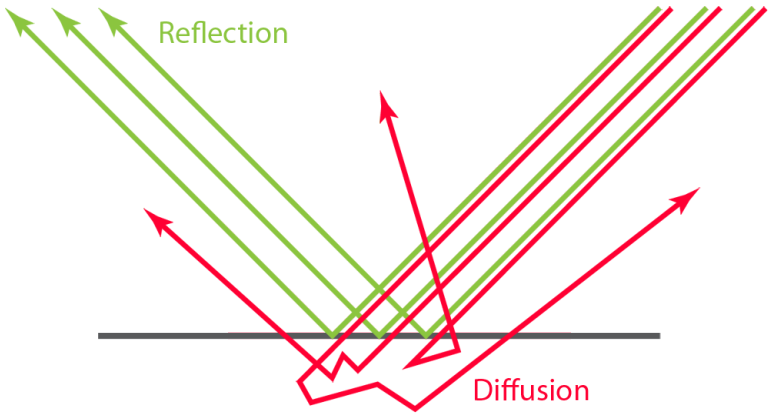

specular vs metalness
specular workflow
metalness workflow
albedo map
specular map
metalness map
albedo map

Conservation of Energy

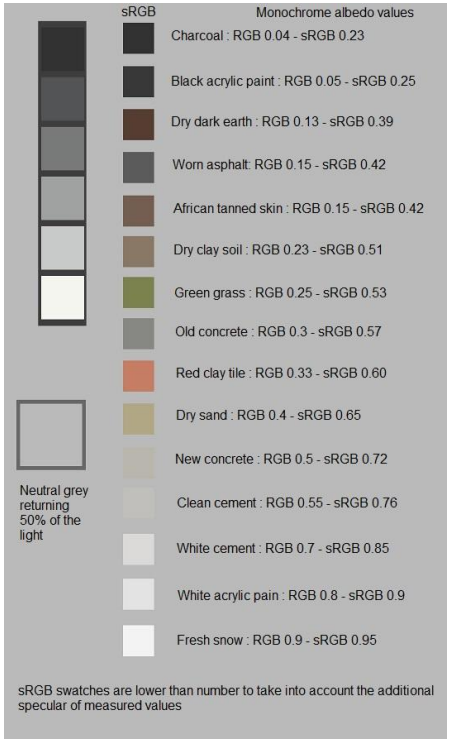
- An object can not reflect more light than it receives
- Reflection and diffusion are mutually exclusive
  - highly reflective = little to no diffuse light
  - bright diffusion = little to no reflection

Diffuse/Albedo

- Absorption
- Scattering
- Color to describe the fractions of light
- Other names
  - Diffuse Light
  - Diffusion
  - Subsurface Scattering



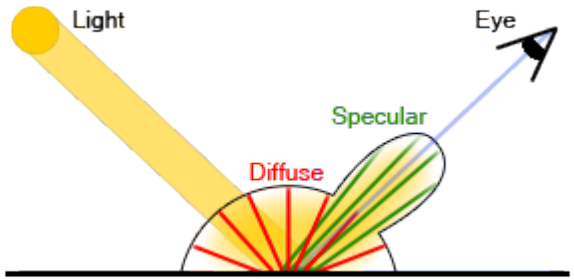
Diffuse Color



- 32-243 in sRGB
- charcoal = darkest
- snow = brightest

Specular/Reflectivity

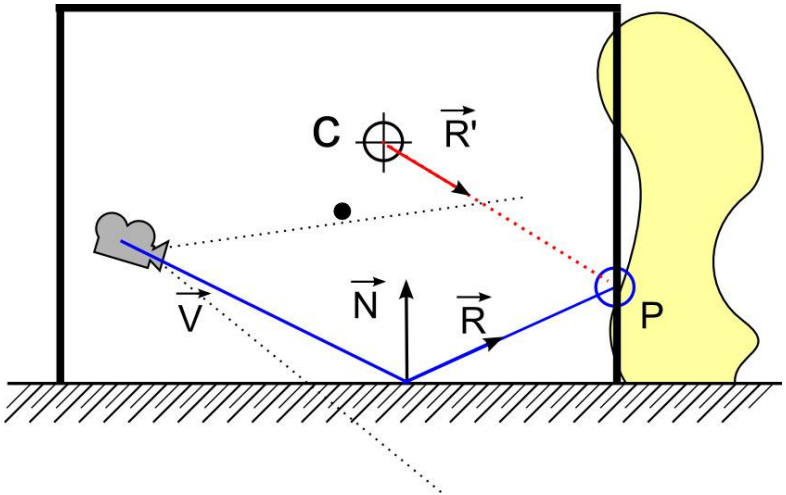
- Light bounces
  - At the opposite angle
- Percentage of light that bounces is called its specular or reflective value
- Two different types of substances
  - Dielectric (0.02 - 0.05)
  - Metallic (0.5+)




EXIT
20:35

Reflection Probe

- Captures a spherical view of its surroundings
- Stored as a cubemap
- Space/time requirement
- Process
  - Get mirror direction (R) from view vector (V) and object's normal (N)
  - Intersect (P) the mirror direction with the scene geometry proxy
  - Retrieve lighting information for this point based on position of the cubemap (C) and the surface's roughness (R`)



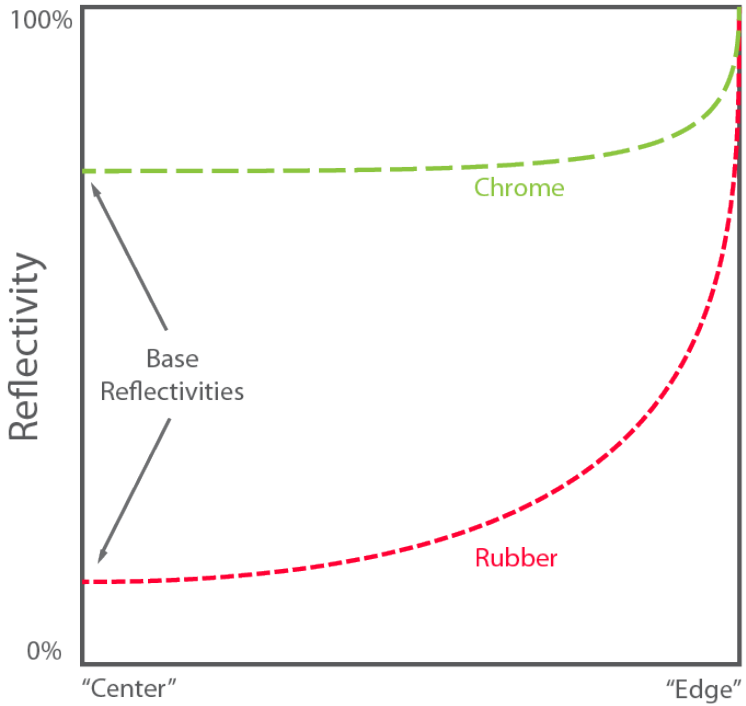
Reflection Probe

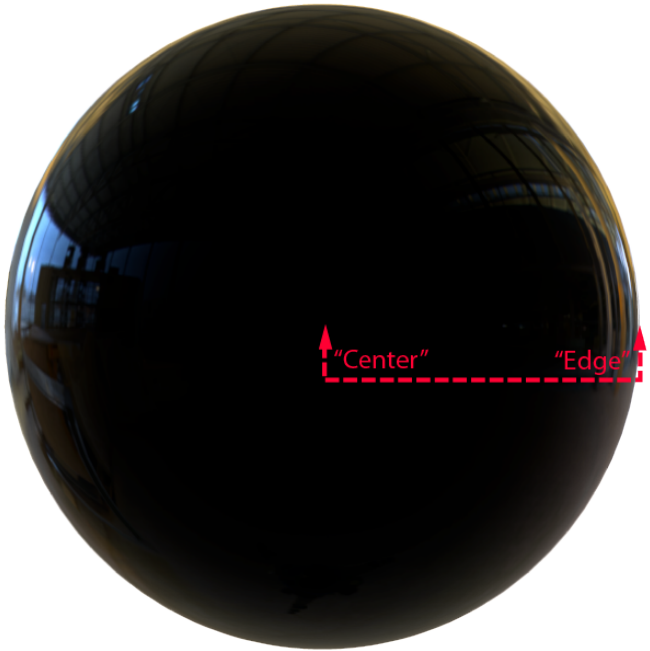
Remember Me (2013)

Fresnel

- Materials have higher reflectivity at “grazing angles”
- Everything has Fresnel
- Already calculated in PBR
- Consults material’s base reflectivity and glossiness

Fresnel

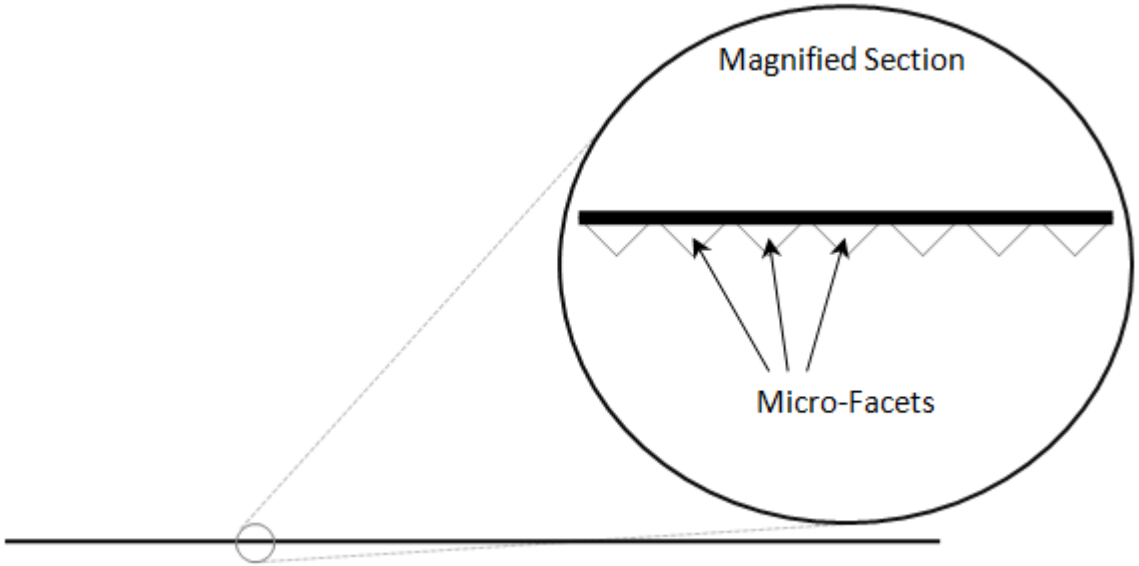

100%
Reflectivity
0%
Chrome
Base
Reflectivities
Rubber
"Center"
"Edge"


"Center"
"Edge"

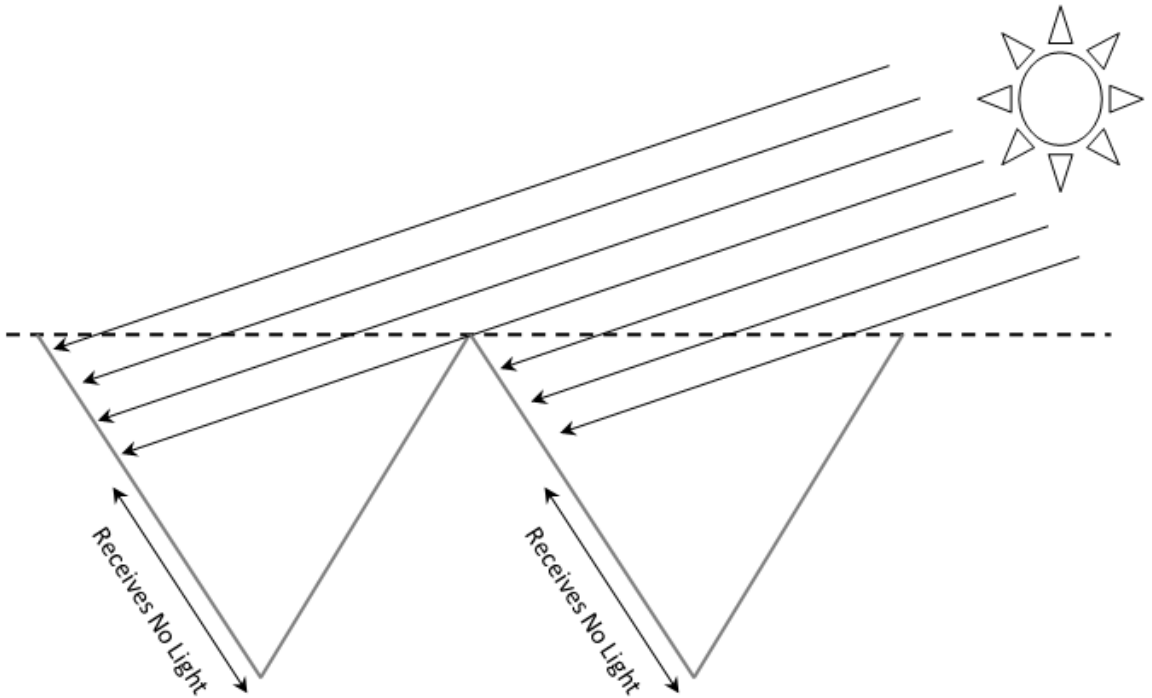
Roughness

- Small imperfections
  - grooves, cracks, lumps
  - scratches, dents
  - self shadows
- Diffusion/Reflection
  - automatically impacted

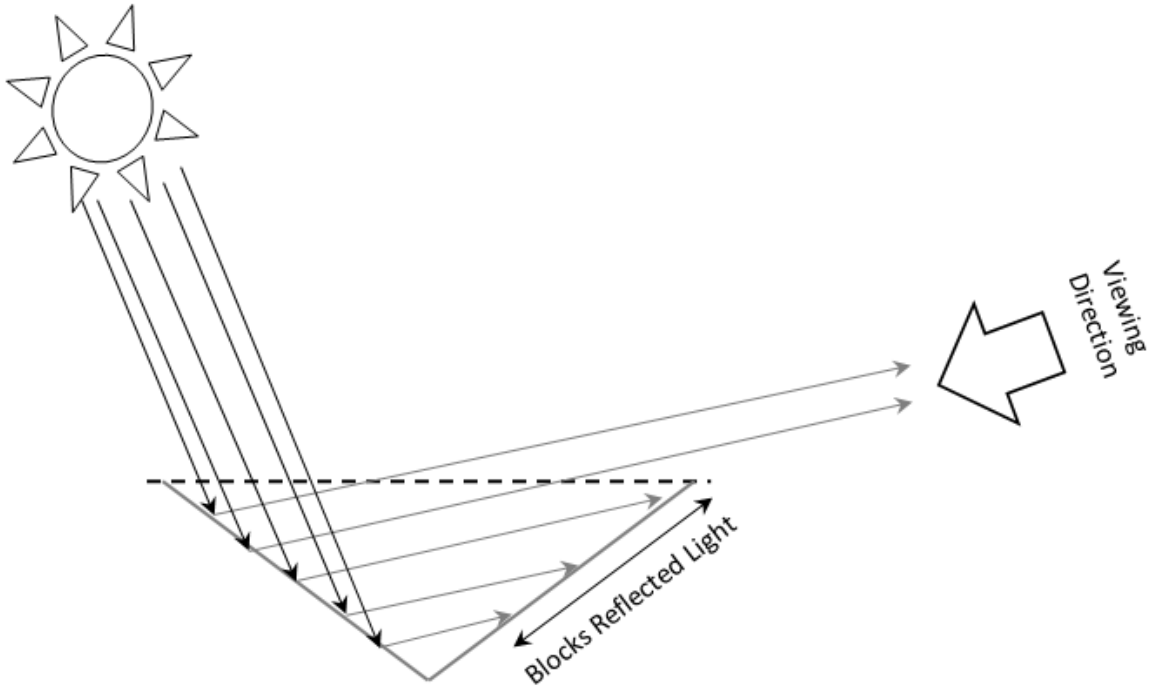
Microfacet



Microfacet Incoming

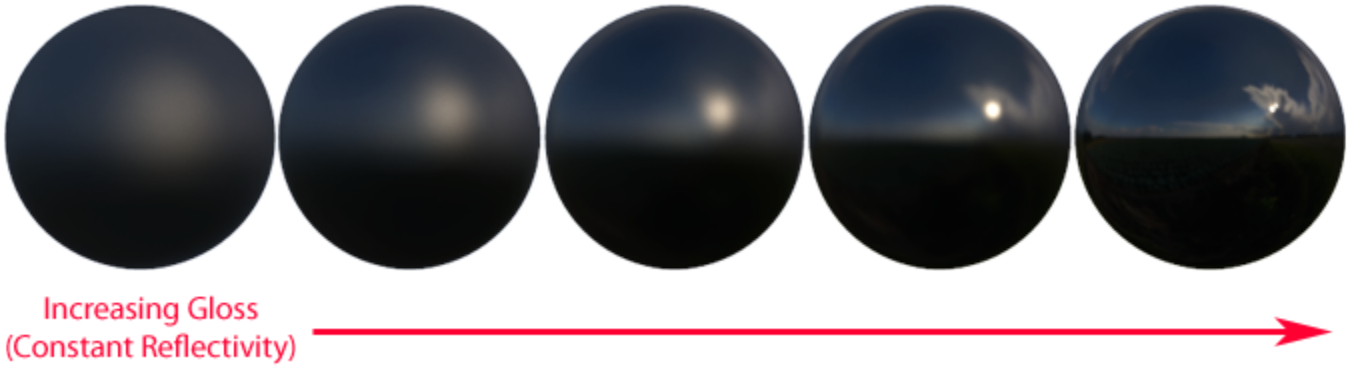


Microfacet Outgoing



Roughness

- Low roughness
  - Very small but bright light
- High roughness
  - Very large but dull light



Roughness

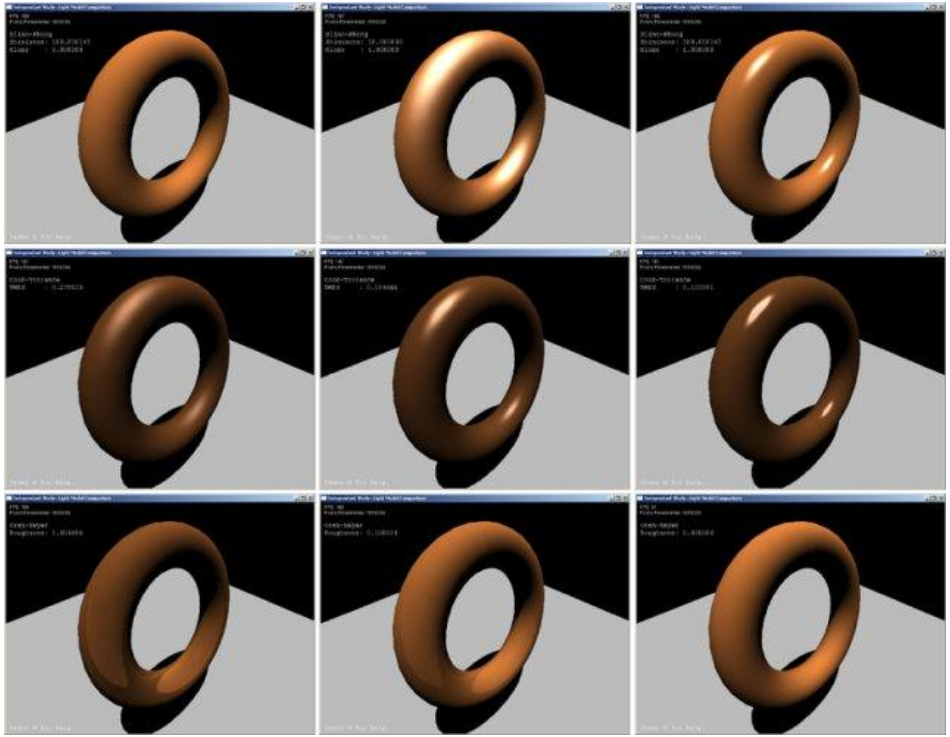
Optional

- Emission
- Ambient Occlusion
- Detail albedo
- Heightmap for parallax

Shading Models

- Old: Blinn-Phong
  - Pro
    - Inexpensive
  - Cons
    - Artificial
    - No regard to conservation of energy
- New: Cook-Torrance
  - Pro
    - Realism
  - Con
    - Expensive

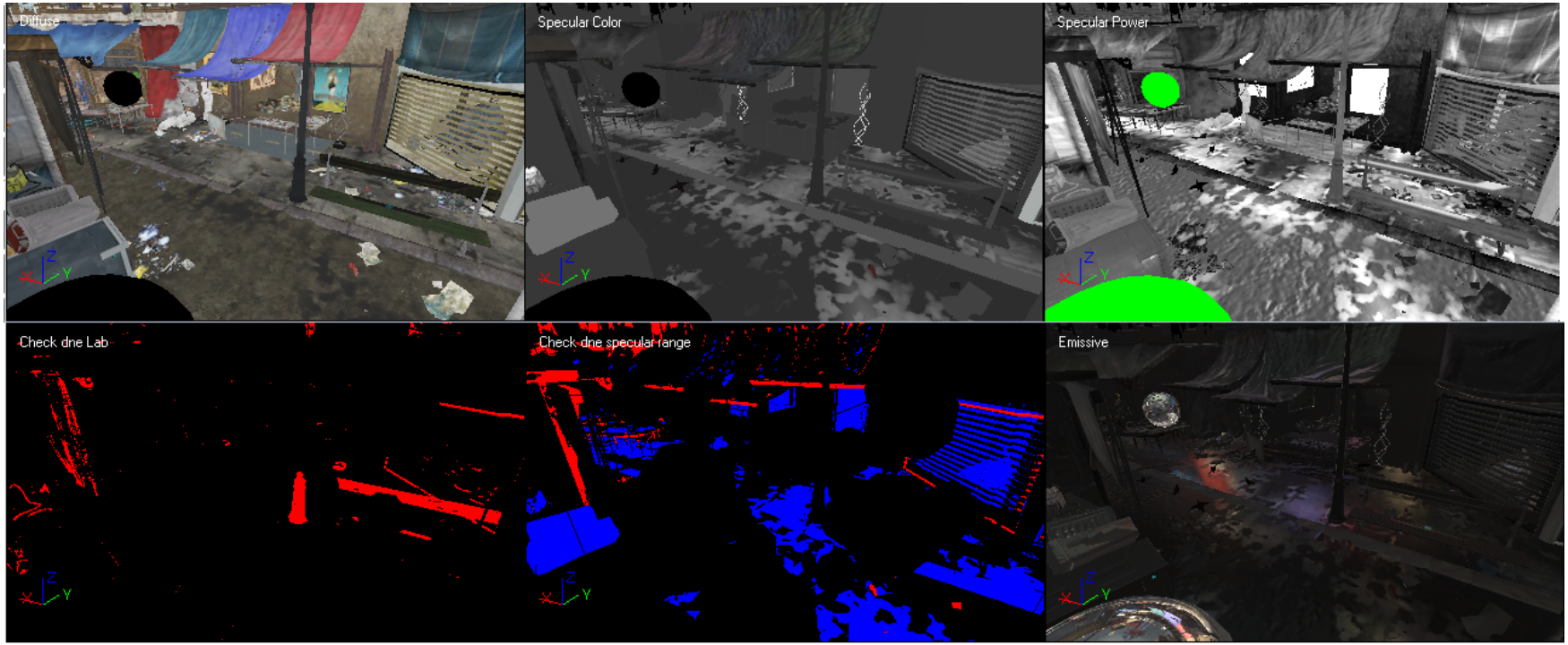
Comparison



Game Engines / Toolkits

- Game Engines
  - UE4
  - Unity 5
- Toolkits
  - Marmoset
  - Disney BRDF Explorer
  - alShaders (arnold)

Extra


Diffuse
Specular Color
Specular Power
Check dne Lab
Check dne specular range
Emissive

Photogrammetry

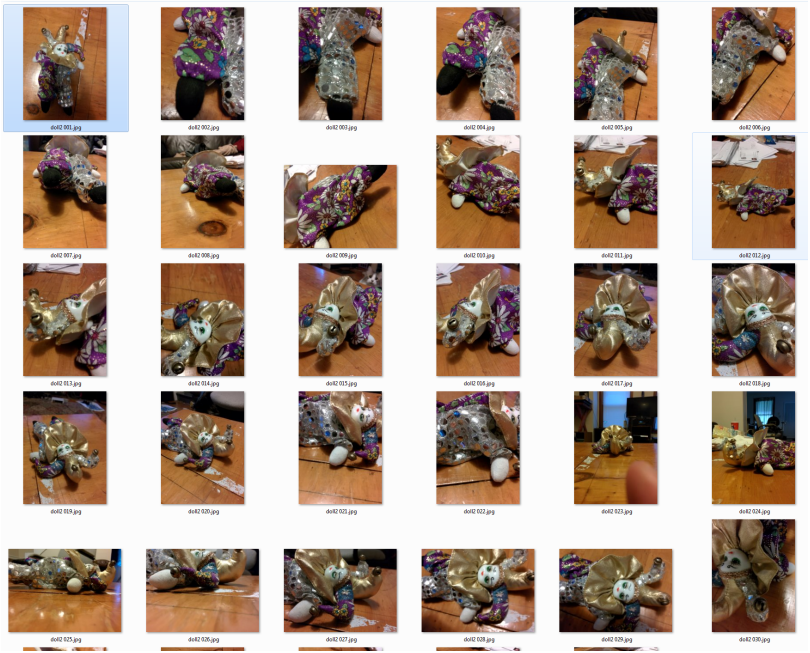
Photogrammetry

- Procedurally generate meshes from real world photographs
- Has the potential to significantly reduce art budgets, or to hire on clay modellers instead of digital modellers
- Used in The Vanishing of Ethan Carter, Kite Demo by Epic, Konami’s Fox Engine

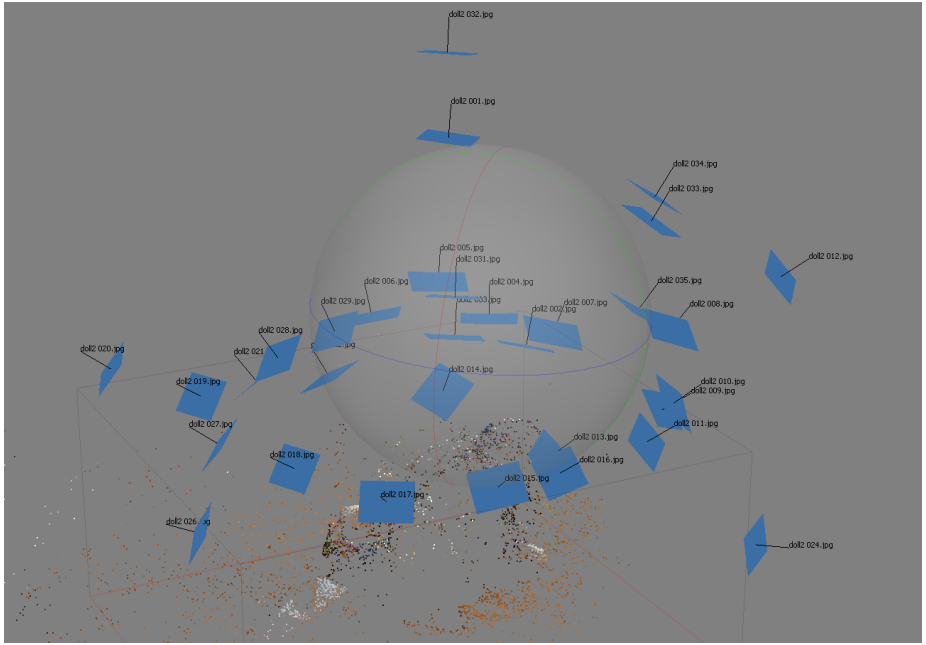
Photogrammetry

- Epic’s Kite Demo used Agisoft to create more than 250 high quality assets

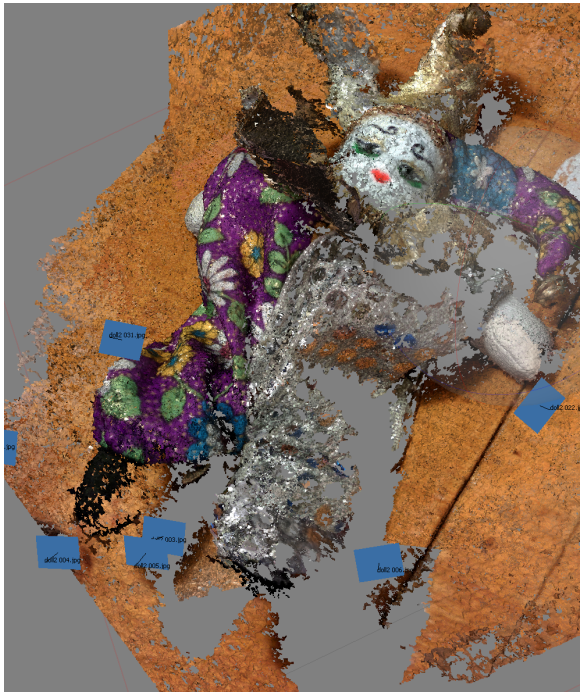
Agisoft Workflow



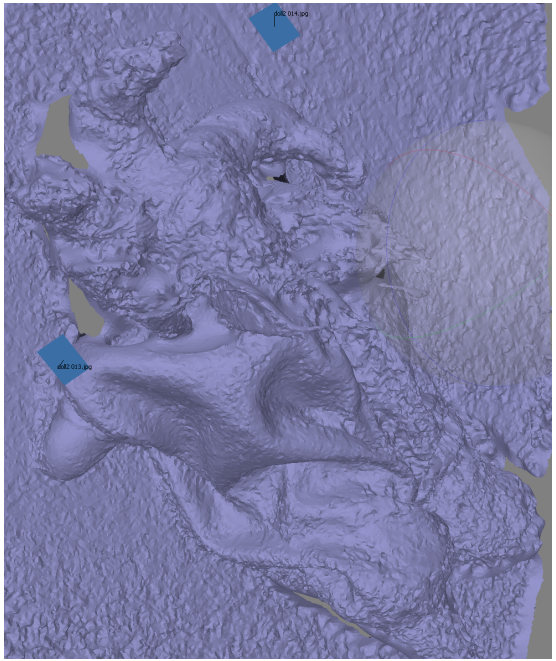
Agisoft Workflow



Agisoft Workflow



Agisoft Workflow



Agisoft Workflow

Agisoft Complications

- Demo version doesn’t export meshes
- Meshes are too high poly to be used immediately (doll was 274,000 faces)
- License cost $179 for basic version and $3499 for full version (Educational versions were available for reduced costs)
- Shadows are baked into texture

Photogrammetry Notes

- Not a full replacement for artists, but could be a potential time saver
- The baked lighting issue can be procedurally fixed, but requires additional set up
- Photogrammetry isn’t a well established market yet, so business/marketing savvy opportunists could make money

Questions

References

- http://www.marmoset.co/toolbag/learn/pbr-theory
- http://content.gpwiki.org/D3DBook:%28Lighting%29_Cook-Torrance
- http://blogs.unity3d.com/2014/10/29/physically-based-shading-in-unity-5-a-primer/
- http://blog.selfshadow.com/publications/s2014-shading-course/langlands/s2014_pbs_alshaders_slides.pdf
- https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf
- http://www.disneyanimation.com/technology/brdf.html
- https://www.unrealengine.com/blog/physically-based-shading-in-ue4
- https://docs.google.com/document/d/1Fb9_KgCo0noxROKN4iT8ntTbx913e-t4Wc2nMRWPzNk/edit
- https://www.unrealengine.com/blog/physically-based-shading-on-mobile
- http://aras-p.info/texts/files/201403-GDC_UnityPhysicallyBasedShading_notes.pdf
- http://www.fxguide.com/featured/game-environments-parta-remember-me-rendering
- http://oliverfallows.com/independent_study
- http://en.wikipedia.org/wiki/Photogrammetry
- http://en.wikipedia.org/wiki/Screen_space_ambient_occlusion
- https://www.youtube.com/watch?v=_zVddsYKZnw
- https://www.youtube.com/watch?v=clakekAHQx0
- https://www.youtube.com/watch?v=X-imQZh5568

References

- https://docs.unrealengine.com/latest/INT/Engine/Rendering/LightingAndShadows/DistanceFieldAmbientOcclusion/index.html
- http://www.agisoft.com/
- http://forums.anandtech.com/showthread.php?s=62f6e932ce27fc67e259b8ee256984e6&t=2359964
- http://http.developer.nvidia.com/GPUGems3/gpugems3_ch12.html
- https://docs.unrealengine.com/latest/INT/Engine/Rendering/LightingAndShadows/AmbientOcclusion/index.html