VU Entwurf und Programmierung einer Rendering-Engine

Materials and Lights for Physically Based Rendering

Christian Luksch

Motivation

Unreal Paris, http://www.benoitdereau.com/

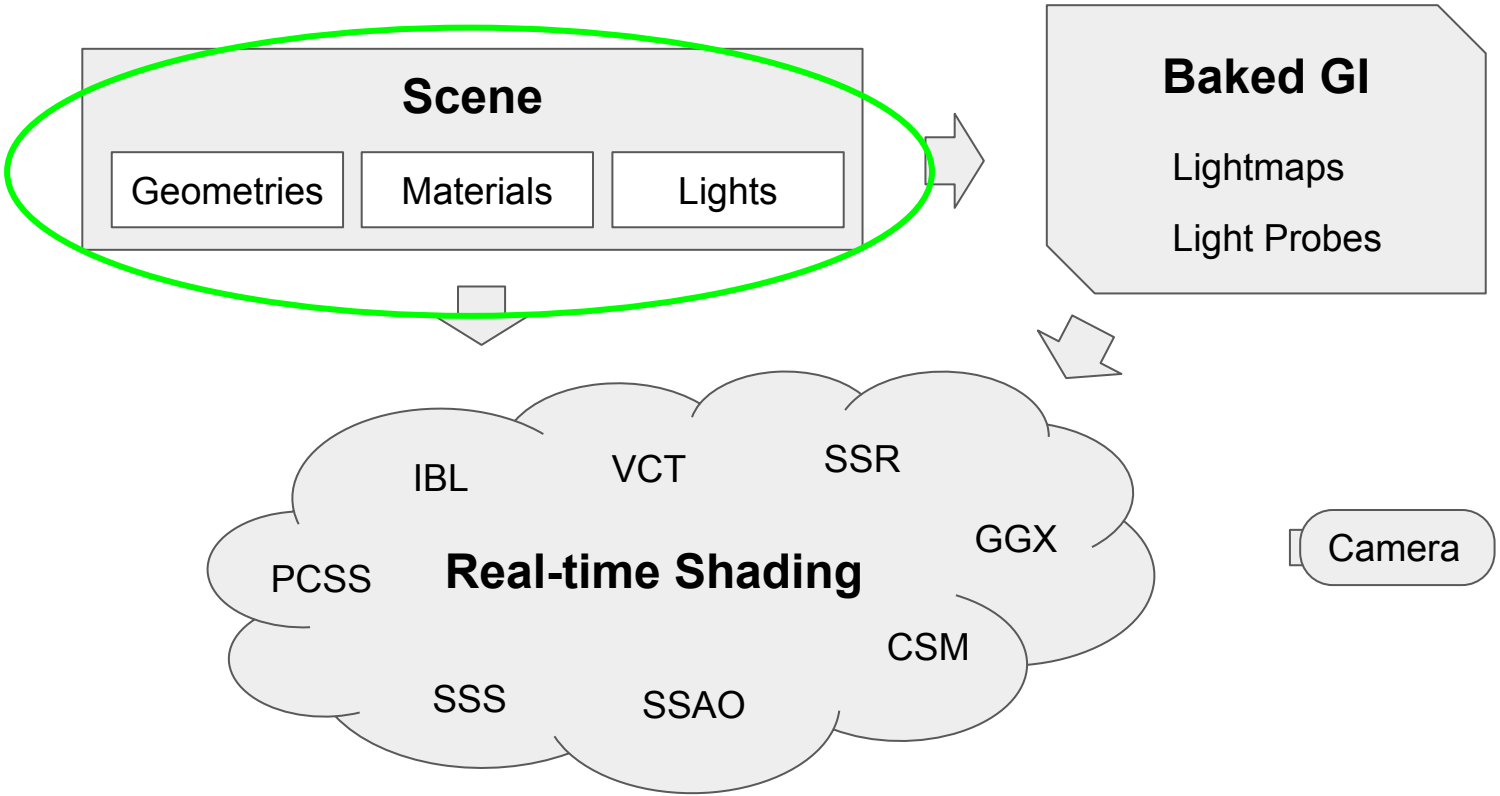
Motivation

GTA V

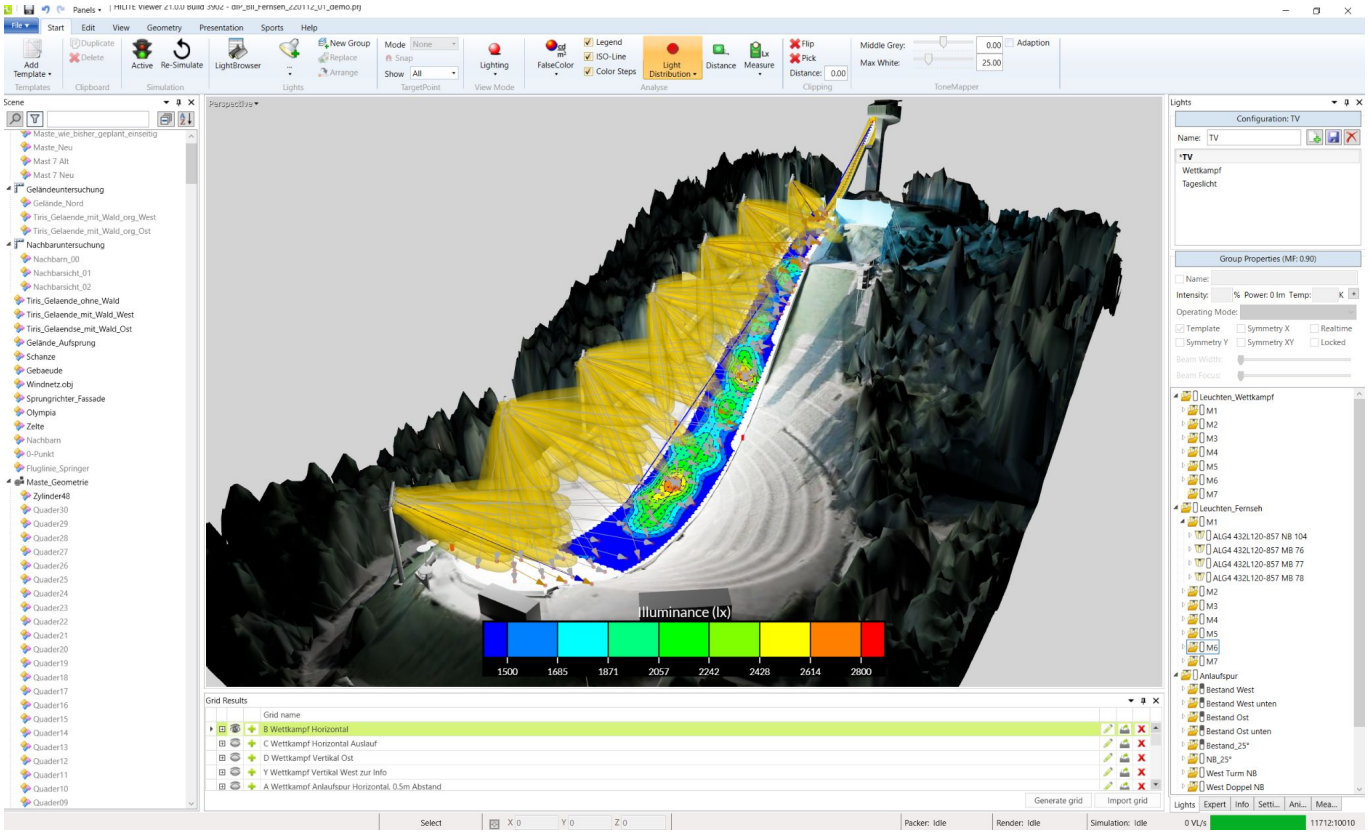
Motivation

Unity, Cities Skyline

Rendering System



Demo: Lighting Design Tool



Individual Draw Code - Aardvark Render SG

```
public ISg CreateSg(IAdaptiveValue<ViewportValue> vp)
{
    var mesh = CameraMeshGeometry;

    const double MinDist = 0.4;
    const double MaxDist = 1.0;

    var blendFactor =
        vp.Map(Trafo, (v, t) =>
        {
            var dist = (v.View.GetViewPosition() - t.GetModelOrigin()).Length;

            if (dist > MaxDist) return 1.0;
            if (dist < MinDist) return 0.0;

            return (dist - MinDist) / (MaxDist - MinDist);
        });

    return mesh.ToSg()
               .Trafo(Trafo)
               .Uniform("Color", blendFactor.Map(x => new C4f(0.2, 0.2, 0.2, x)))
               .BlendMode(blendFactor.Map(x => x < MaxDist ? BlendMode.Blend : BlendMode.None))
               .DepthWrite(blendFactor.Map(x => x > MinDist))
               .Uniform(HeadLight.Intensity, AdaptiveValue.Constant(0.4f))
               .Surface(HeadLightShader);
}
```

Proposal: Unified RenderObjects

```
type RenderObject_V1 = {
 Mesh : Mesh
 Material : Material
 Transform : Trafo3d
}
```

```
type RenderObject_V2 = {
 Mesh : Mesh
 Material : Material
  * AlphaBlend : aval<bool>
  * DoubleSided : aval<bool>
  * HasGlossyRefl : aval<bool>
 Transform : Trafo3d
 Bound : Box3d
 IsVisible : aval<bool>
 CastShadow : aval<bool>
 ReceiveLight : aval<bool>
}
```

Engine Magic

Missing:

- Custom Shaders
- Text
- …

Individual Draw Code vs Unified RenderObjects

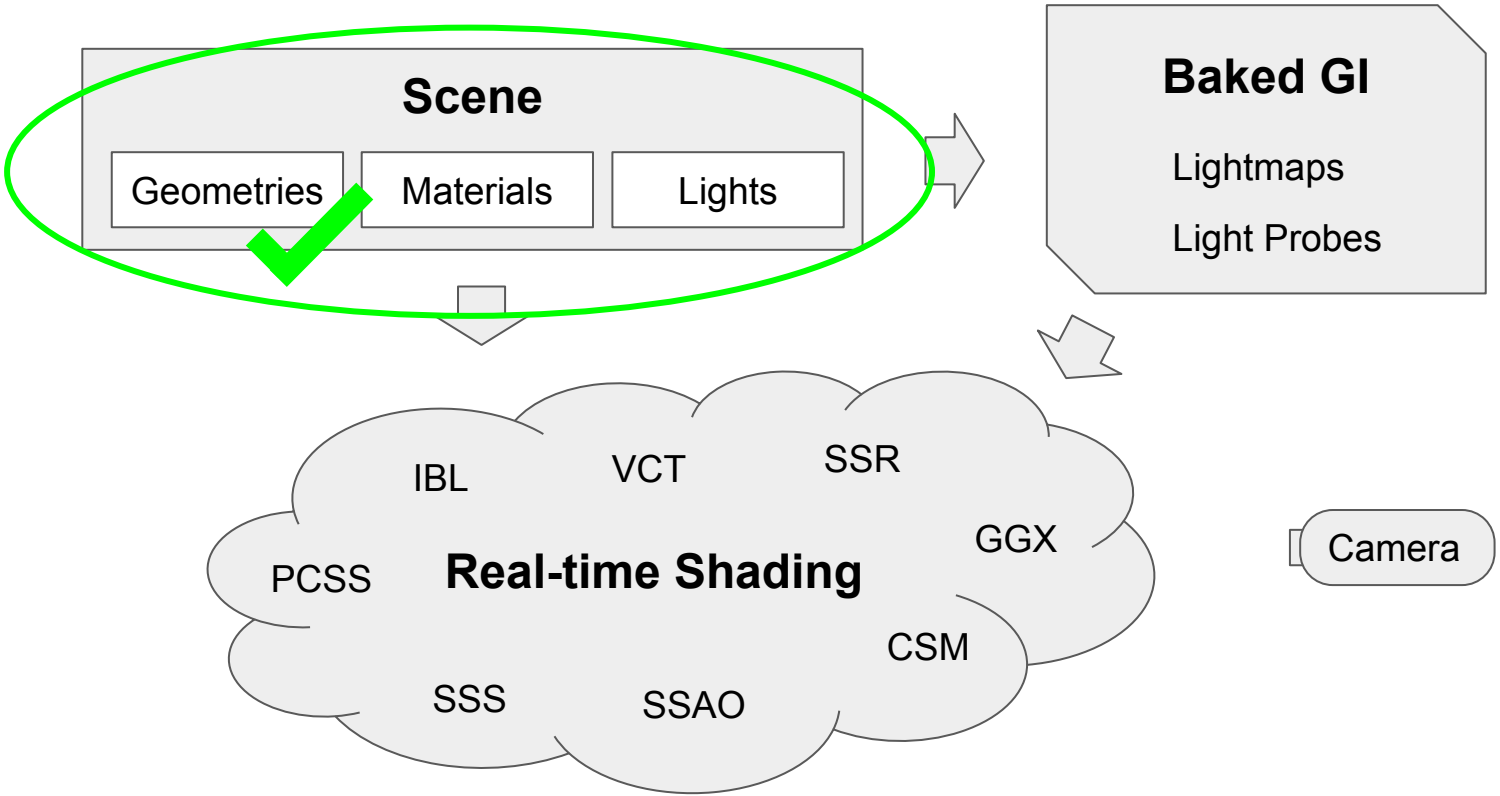
Pros:

- **Performance**: possibility to pack, sort, compute, …
- **Extensibility**: render styles, panorama/stereo output
- **Composition**: multi-pass rendering, draw order / transparency
- **Mainability**: unified processing, less code

-> Handle other aspects (e.g. Transform-/Targetable) in separate “System”

Always some remaining special code? Cons?

Rendering System

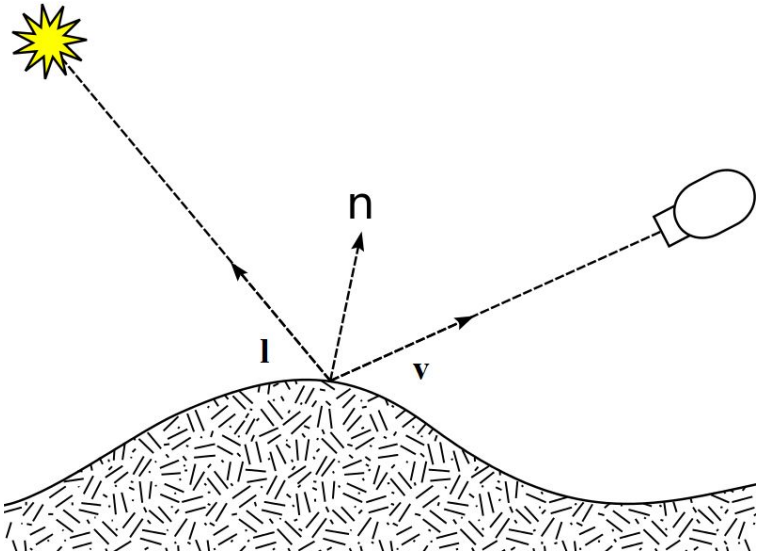


Overview

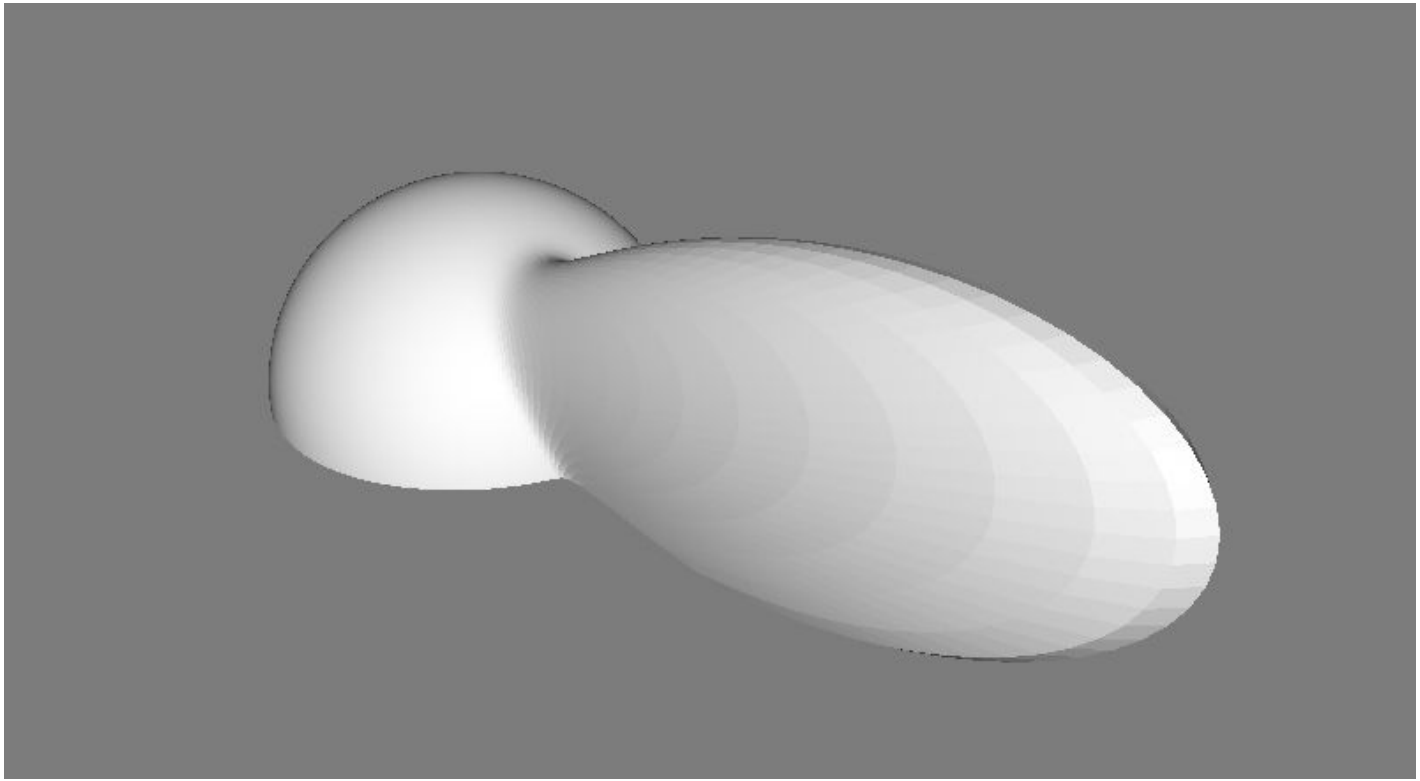
- Scene Content Definition
  - Physically plausible materials
  - Physical light source description
- Putting Everything Together
  - Shading System
- Integration of Global Illumination
  - Baking
  - Real-time

Need for a BRDF

- Bidirectional Reflectance Distribution Function
  - Gives ratio of incoming to outgoing radiance over the hemisphere
  - Unit: sr^{-1} (solid angle)
- Short: $f(\mathbf{l}, \mathbf{v})$



Need for a BRDF



BRDF in Rendering Equation

- Sum up all incoming radiance (evaluated recursively)
  - weighted by projected area
  - weighted by BRDF

$$L_o(\omega_o) = L_e(\omega_o) + \int_\Omega f_r(\omega_o, \omega_i) L_i(\omega_i)(\omega_i \cdot n) d\omega_i$$

Requirements for Material Model

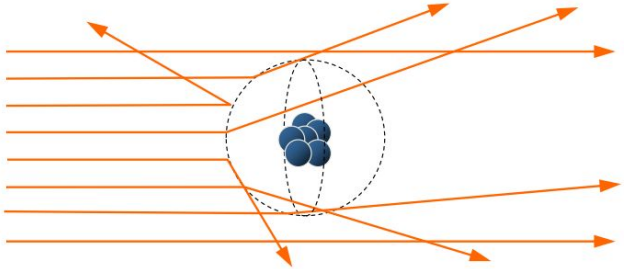
- Physically plausible
  - Energy conservation
  - Reciprocity
- Flexibility
  - Wide range of materials
- Easy to use
  - Meaningful parameters
  - Not possible to violate energy conservation

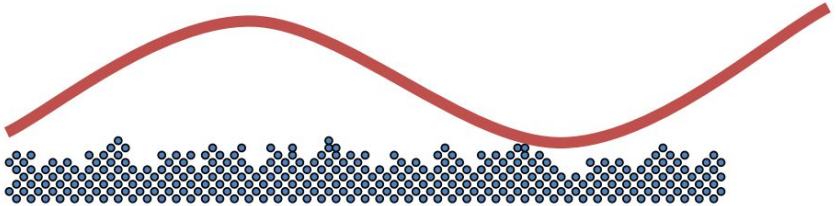
https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/PhysicallyBased/index.html

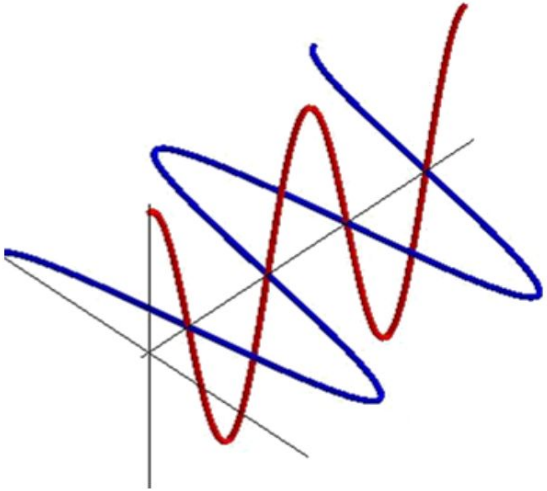
What BRDF Model?

- Phong (1975)
- Blinn-Phong (1977)
- Cook-Torrance (1981)
- Ward (1992)
- Oren-Nayar (1995)
- Lafortune (1997)
- **GGX (2007)**
- Kurt (2010)
- SGD (2012)
- ...

The Physics Behind Shading

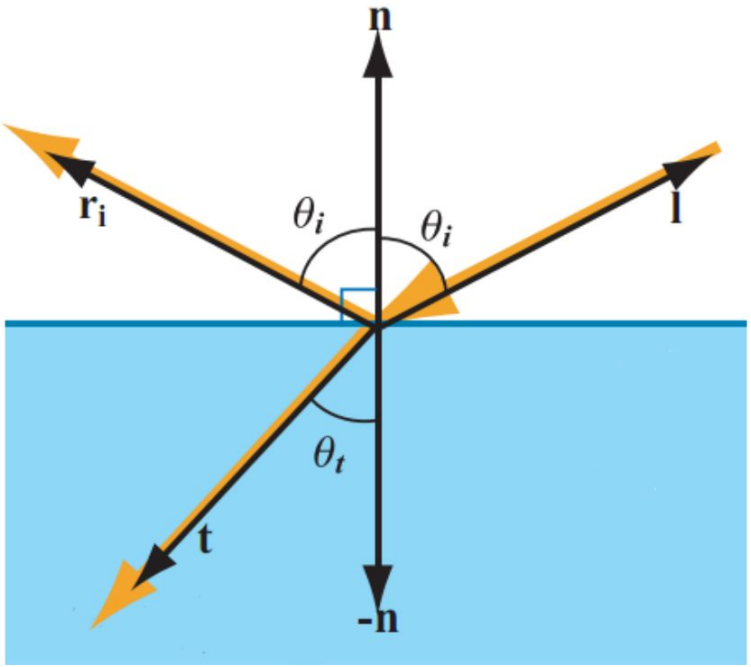




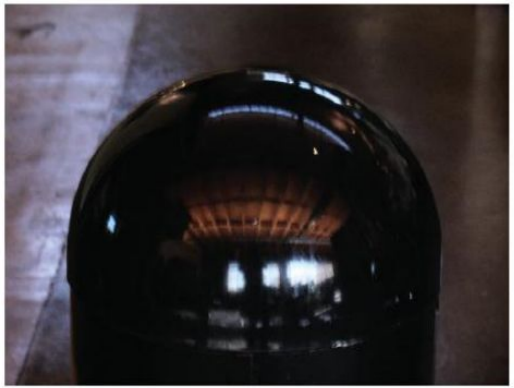


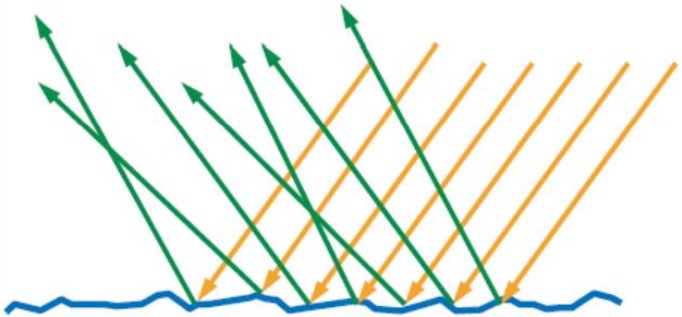
Reflection & Refraction

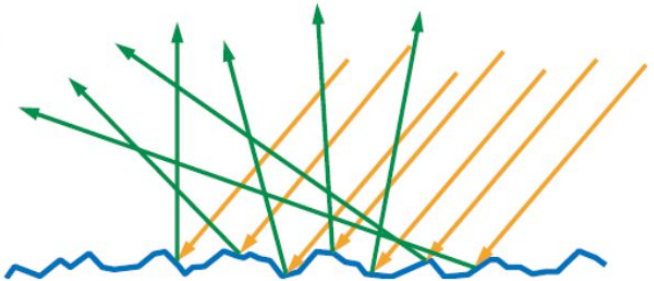
- Fresnel Equations
  - Ratio of reflected and refracted light
- Law of Reflection
  - Angle of incidence = angle of reflection
- Snell's Law
  - Ratio of sines of incidence and refraction direction = ratio of refractive indices



Microgeometry – Microfacets

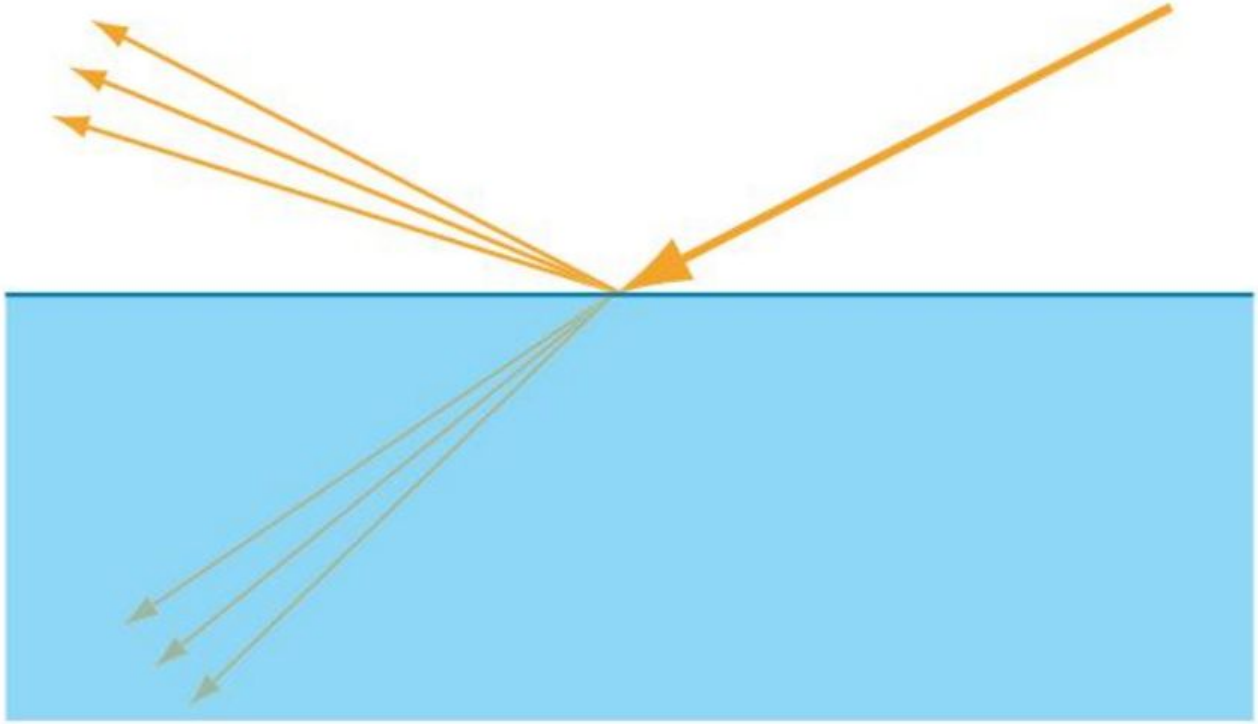






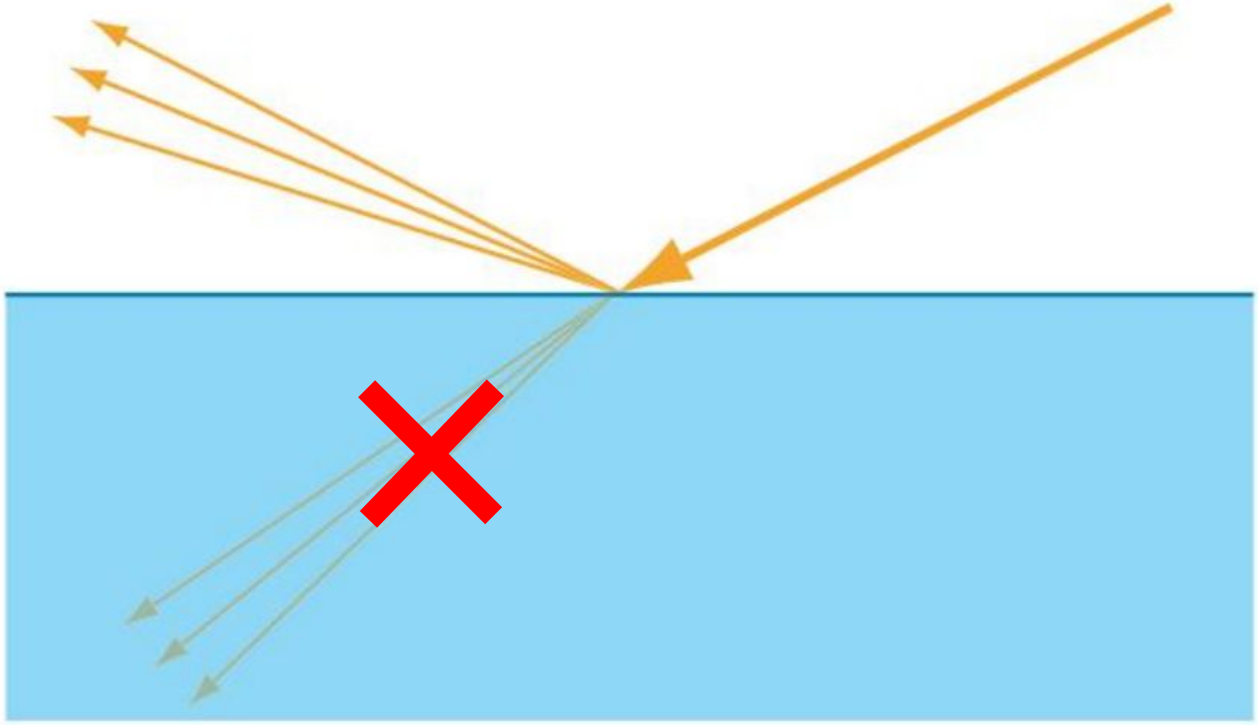
Physics and Math of Shading, Naty Hoffman, SIGGRAPH Physically Based Shading Course 2010

Macroscopic View



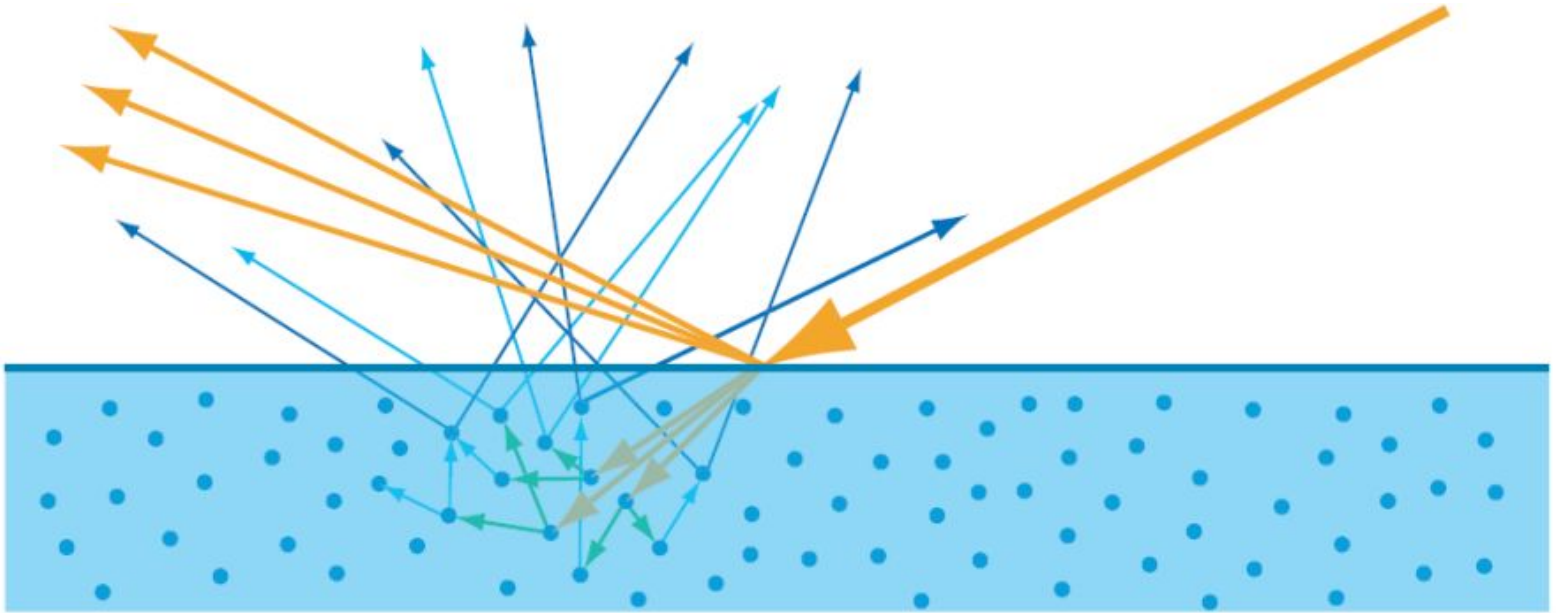
Physics and Math of Shading, Naty Hoffman, SIGGRAPH Physically Based Shading Course 2010

Metals



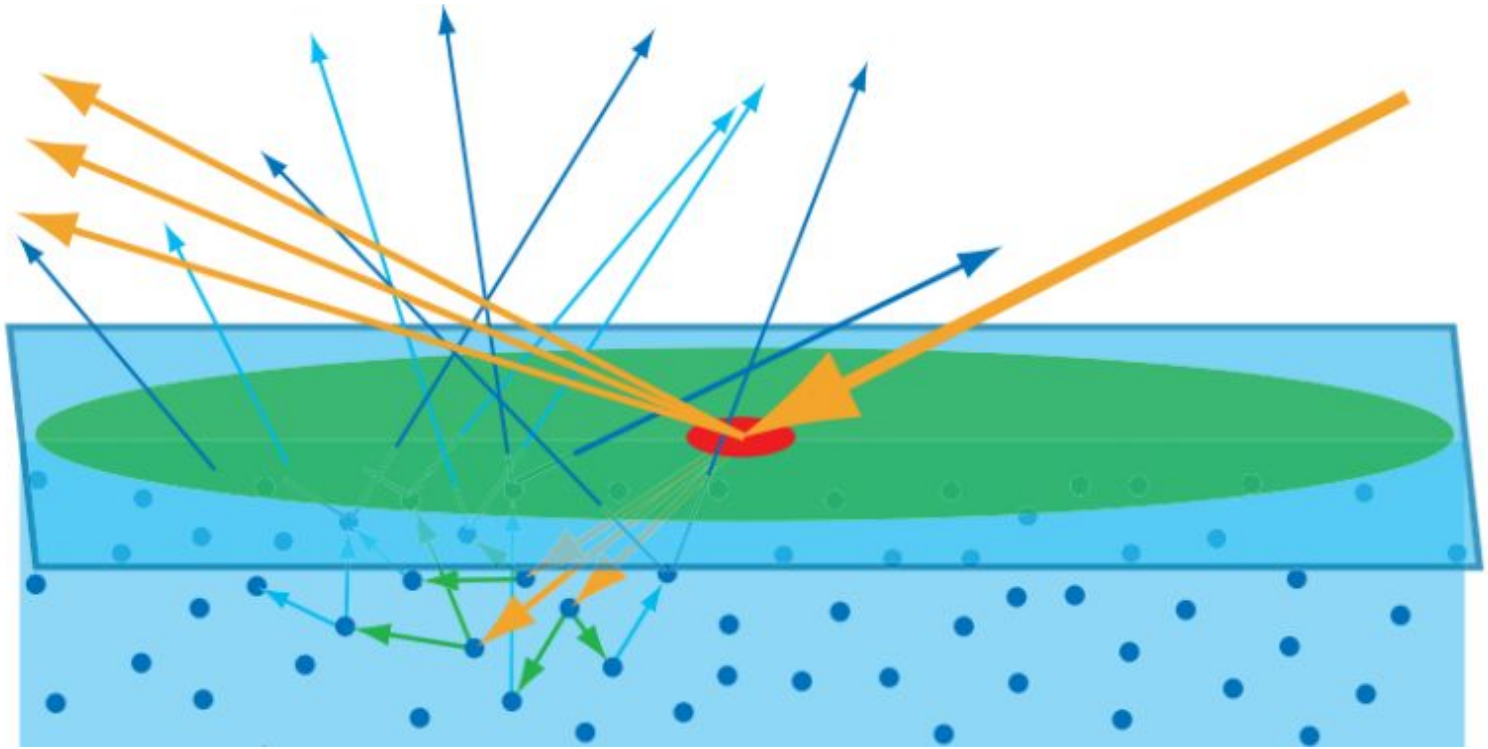
Physics and Math of Shading, Naty Hoffman, SIGGRAPH Physically Based Shading Course 2010

Non-Metals (Insulators, Dielectric)



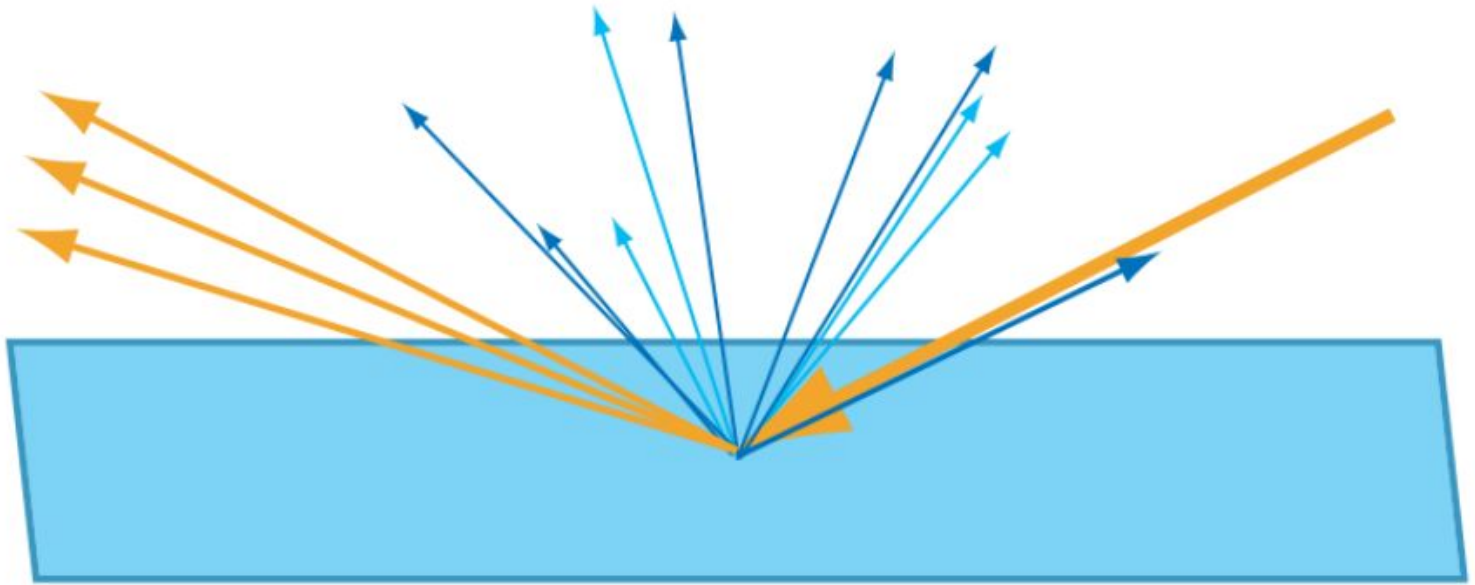
Physics and Math of Shading, Naty Hoffman, SIGGRAPH Physically Based Shading Course 2010

Subsurface Scattering & Scale



Physics and Math of Shading, Naty Hoffman, SIGGRAPH Physically Based Shading Course 2010

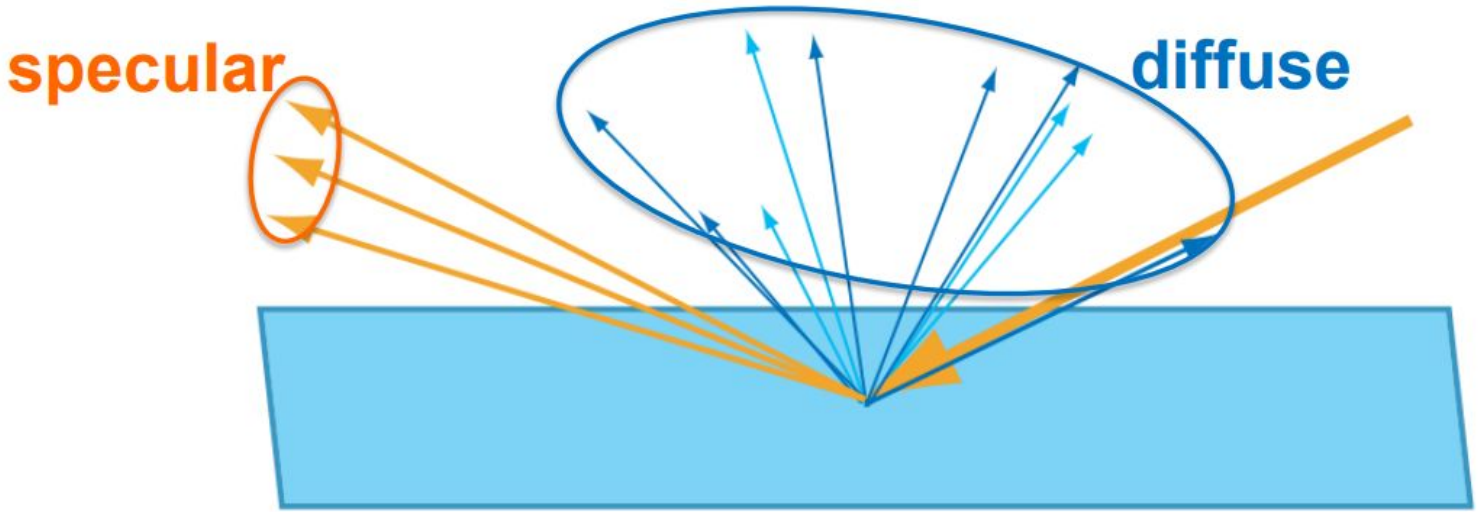
Subsurface Scattering & Scale



Physics and Math of Shading, Naty Hoffman, SIGGRAPH Physically Based Shading Course 2010

Shading Terms

- Ignore entry-exit distance and shade at single point
- Model reflection as “specular” and scattering as “diffuse” separately



Deriving a BRDF

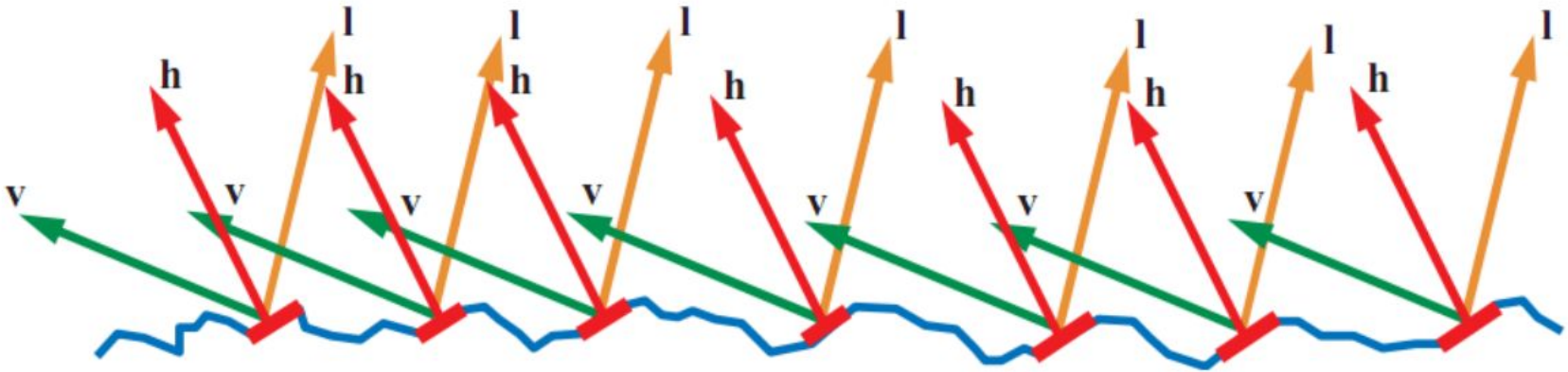
- Apply microfacet theory for rough surfaces
- Apply fresnel equations
- Define combination of “diffuse” and “specular”
- Validate against measured materials

MERL BRDF Database: A Data-Driven Reflectance Model
Matusik et al. 2003, https://www.merl.com/brdf/

EPFL BRDF Database: An Adaptive Parameterization for Efficient Material Acquisition and Rendering
Dupuy and Jakob 2018, https://rgl.epfl.ch/materials

Microfacet Theory

- Facets with normal exactly “halfway” between **l** and **v** will reflect light toward the camera



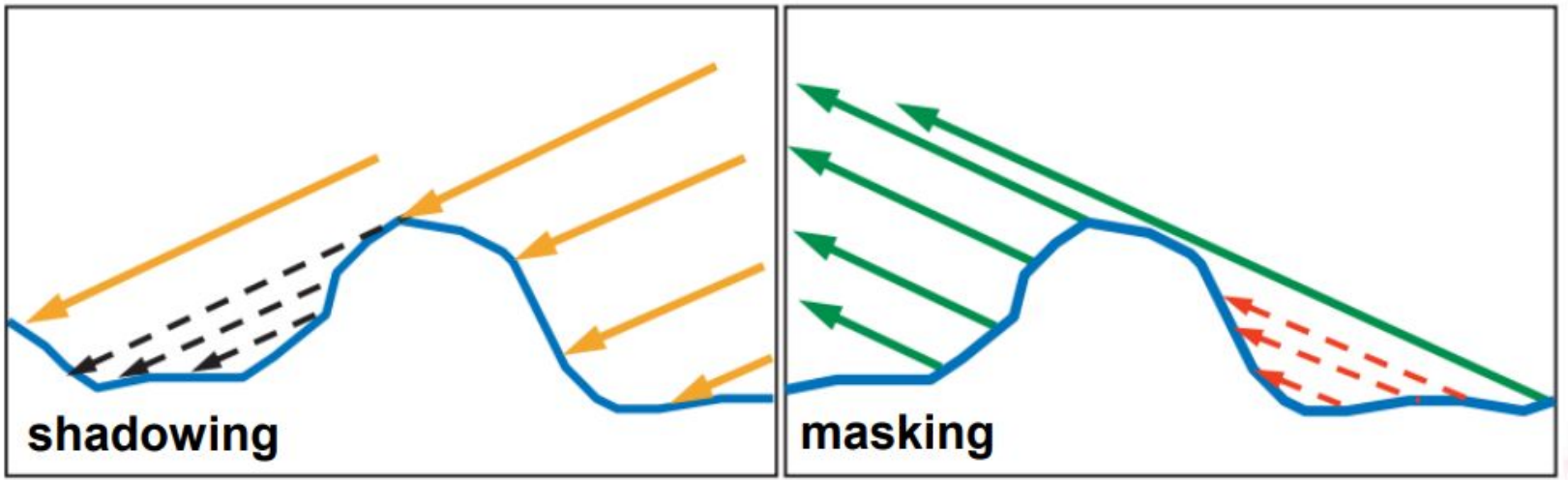
Physics and Math of Shading, Naty Hoffman, SIGGRAPH Physically Based Shading Course 2010

Half-Vector Based BRDFs → Elongated Reflections

Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008; photographer: Elan Ruskin

Microfacet Theory

- Not all facets receive light -> shadowing
- Not all facets are visible -> masking



Microfacet BRDF

- Generalized Cook-Torrance shading model:

$$f(\mathbf{l},\mathbf{v}) = \frac{F(\mathbf{l},\mathbf{h})G(\mathbf{l},\mathbf{v},\mathbf{h})D(\mathbf{h})}{4(\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})}$$

Microfacet Distribution Term

- Also called **N**ormal **D**istribution **F**unction
  - Gives probability of a microfacet oriented in a certain half-vector direction
  - Microfacet normal **h** sometimes also referred to as **m**
- Defines shape/falloff of specular highlight
  - Includes user parameter “Roughness”
- Normalized such as:

$$\int_{\Omega} D(h)\,(n \cdot h)\,d\omega_h = 1$$

Different NDFs

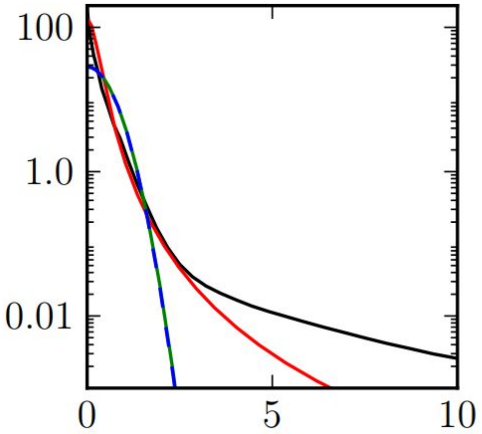
- Blinn-Phong: $D_{Phong}(h) = \frac{power + 2}{2\pi}(n \cdot h)^{power}$
- Beckmann: $D_{Beckmann}(h) = \frac{1}{4\pi\alpha^2(n\cdot h)^4} \exp\left(\frac{(n\cdot h)^2 - 1}{\alpha^2(n\cdot h)^2}\right)$
- Trowbridge-Reitz (X in GGX): $D_{GGX}(h) = \frac{\alpha^2}{\pi\left((n\cdot h)^2(\alpha^2 - 1) + 1\right)^2}$
- Other: Exponential, ABC, SGD, ...

Roughness

https://www.unrealengine.com/en-US/blog/physically-based-shading-in-ue4

Comparison

- Left: Log-scale plots of specular peak vs θ_h (degrees)
  - black: chrome, green-blue: Beckmann/Blinn-Phong, red: GGX
- Right: point-light response from chrome

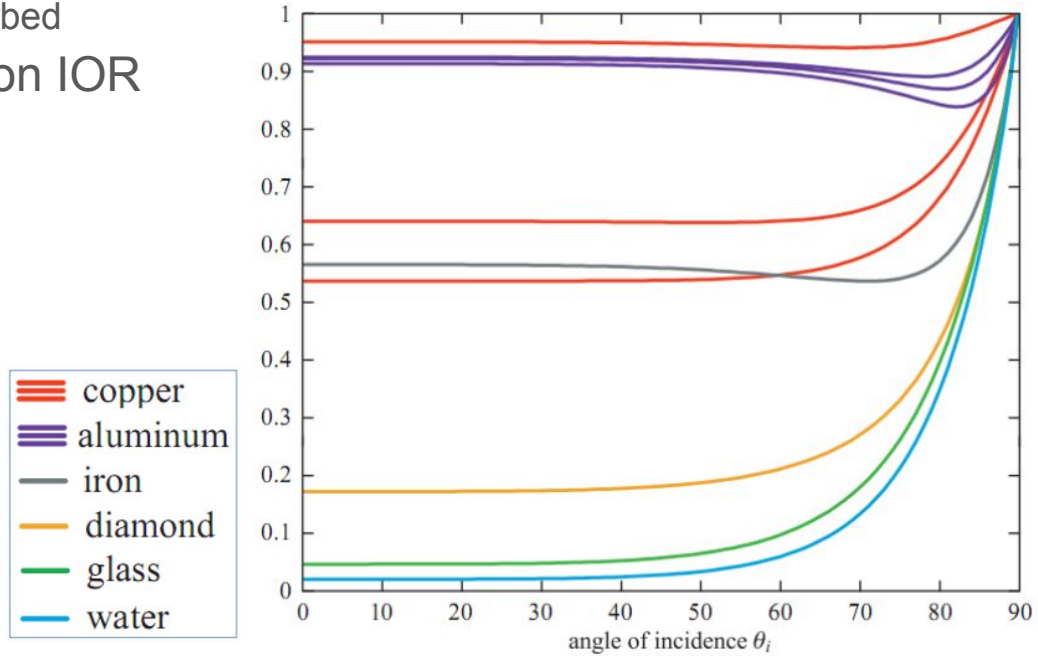


Microfacet Distribution

- Beckmann and Blinn-Phong quite similar (for low roughness)
  - Sharp specular highlight
  - Plausible for a range of materials
  - Used in various BRDFs (Ward, Cook-Torrance, Kurt, ...)
- GGX shows longer “tail”
  - Long tail found in lots of materials
  - Suited for high variety of materials
  - Currently preferred distribution

Fresnel Term

- Defines fraction of light reflected
  - Remaining part refracted/absorbed
- Depends on index of refraction IOR
- IOR depends on wavelength
  - Strong distinction in metals

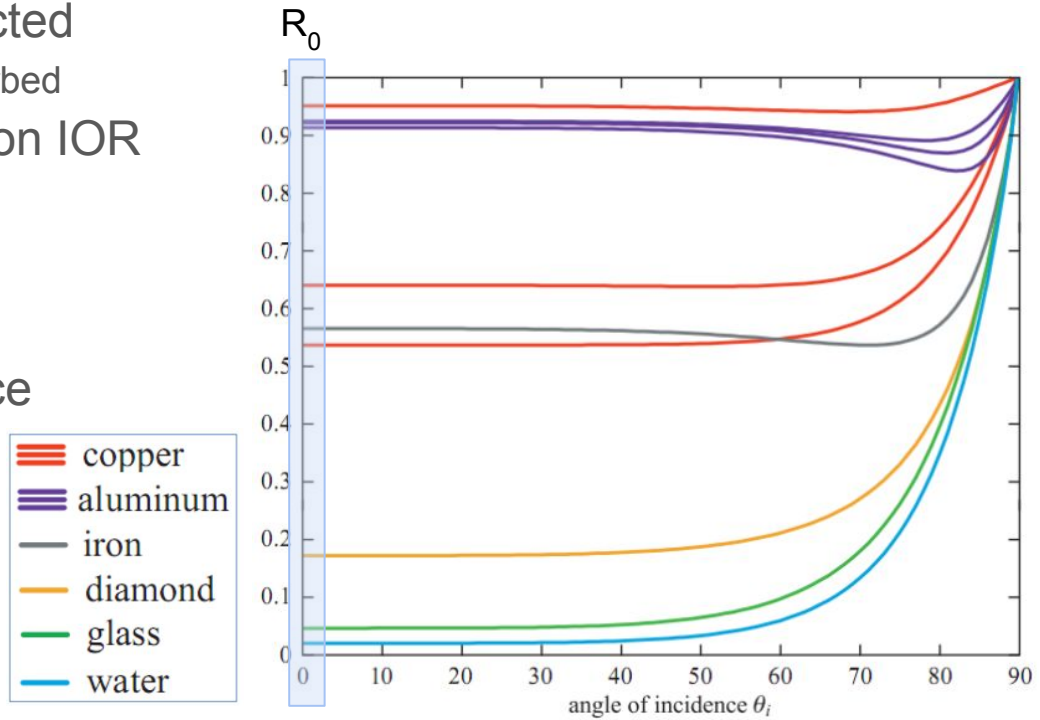


Fresnel Term

- Defines fraction of light reflected
  - Remaining part refracted/absorbed
- Depends on index of refraction IOR
- IOR depends on wavelength
  - Strong distinction in metals

- Reflection at normal incidence

$$R_0 = \frac{(n_1 - n_2)^2}{(n_1 + n_2)^2}$$



Fresnel Term

- Schlick Approximation:

$$F(\mathbf{n}, \mathbf{l}) = R_0 + (1 - R_0)\,(1 - \mathbf{n}\cdot\mathbf{l})^5$$

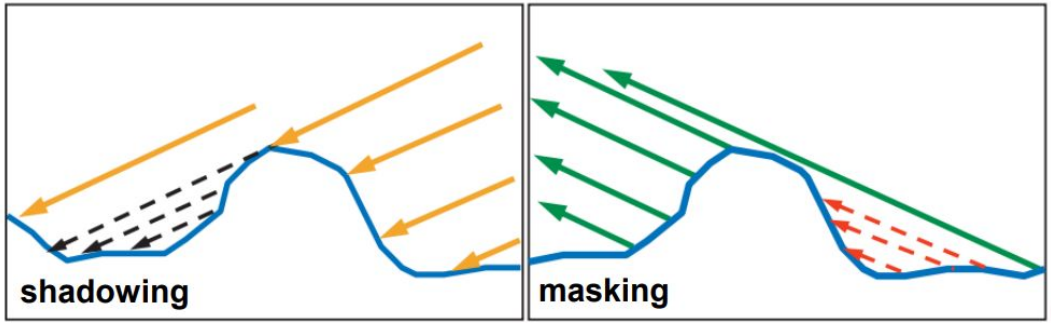
  - Substitute **n** by **h** when applied in microfacet BRDF

- R_0 scalar for non-metals, RGB for metals

Geometry Term

- Gives probability of microfacet being shadowed or masked

$$V(\mathbf{l},\mathbf{v}) = \frac{G(\mathbf{l},\mathbf{v},\mathbf{h})}{(\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})}$$



- Combined with normalization also called Visibility Term $V(l, v)$
  - Factors in denominator usually canceled
- Observations:
  - Depending on microfacet distribution
  - Shadow/Masking symmetrically

Variety of Geometry Terms

- Implicit V(l, v) = 1
- Approximations
  - Cook-Torrance
  - Kelemen-Szirmay-Kalos

$$G_{\text{implicit}}(\mathbf{l_c}, \mathbf{v}, \mathbf{h}) = (\mathbf{n} \cdot \mathbf{l_c})(\mathbf{n} \cdot \mathbf{v})$$

$$G_{\text{ct}}(\mathbf{l}, \mathbf{v}, \mathbf{h}) = \min\left(1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{v})}{(\mathbf{v} \cdot \mathbf{h})}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{l})}{(\mathbf{v} \cdot \mathbf{h})}\right)$$

$$\frac{G_{\text{ct}}(\mathbf{l}, \mathbf{v}, \mathbf{h})}{(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})} \approx \frac{1}{(\mathbf{l} \cdot \mathbf{h})^2}$$

- Derived from D(h)
  - Smith: product of two functions

$$G_{\text{s}}(\mathbf{l}, \mathbf{v}, \mathbf{h}) = G_{\text{s1}}(\mathbf{l}, \mathbf{h}) G_{\text{s1}}(\mathbf{v}, \mathbf{h})$$

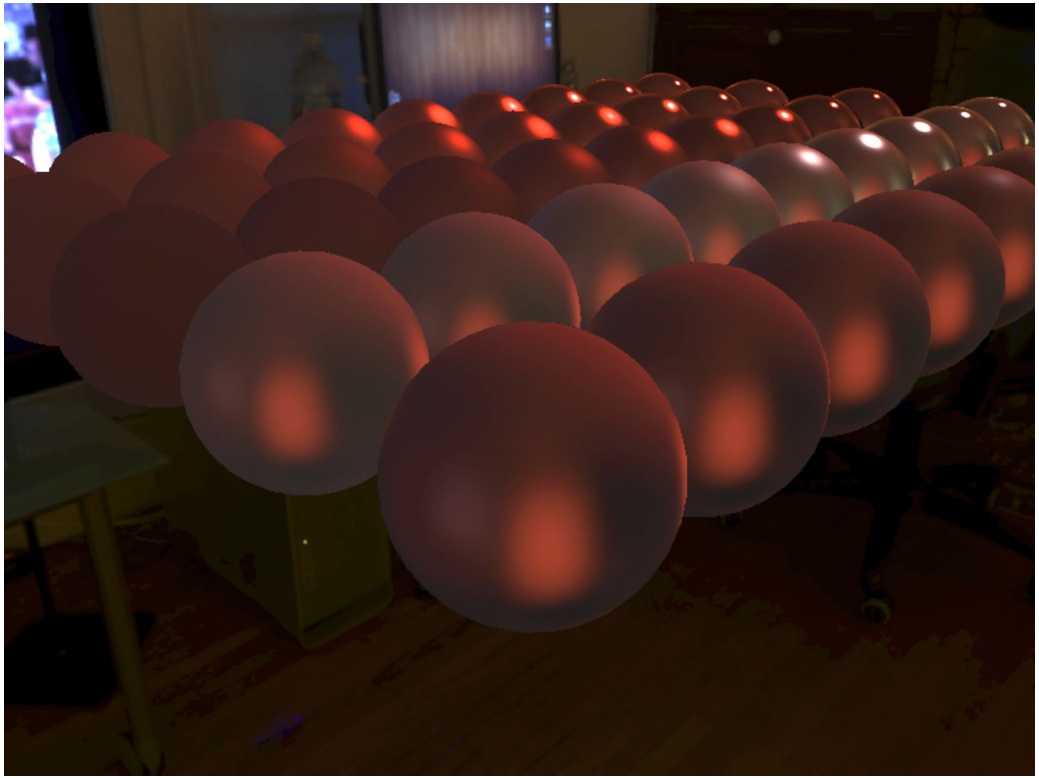
$$\textbf{GGX}: G_1(\mathbf{v}, \mathbf{m}) = \chi^+\left(\frac{\mathbf{v} \cdot \mathbf{m}}{\mathbf{v} \cdot \mathbf{n}}\right) \frac{2}{1 + \sqrt{1 + \alpha_g^2 \tan^2 \theta_v}}$$

$$\textbf{Beckmann}: G_1(\mathbf{v}, \mathbf{m}) \approx \chi^+\left(\frac{\mathbf{v} \cdot \mathbf{m}}{\mathbf{v} \cdot \mathbf{n}}\right) \begin{cases} \frac{3.535a + 2.181a^2}{1 + 2.276a + 2.577a^2} & \text{if } a < 1.6 \\ 1 & \text{otherwise} \end{cases}$$

$$\text{with } a = (\alpha_b \tan \theta_v)^{-1}$$

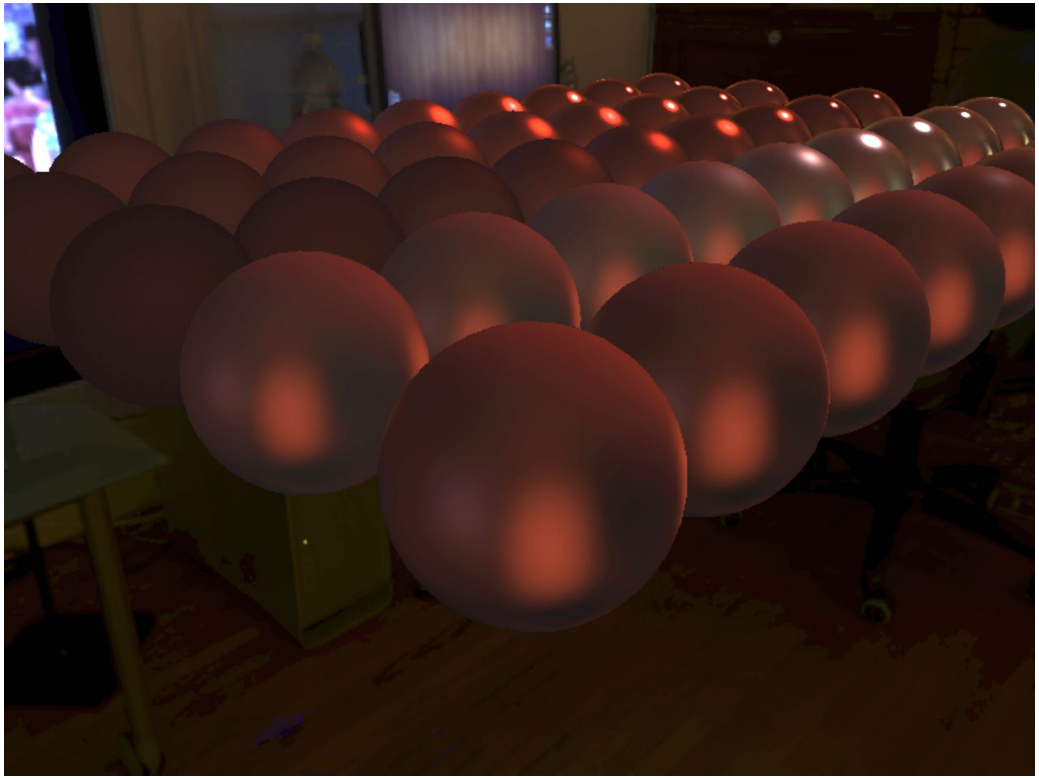
- Independent parameter
  - Measured materials

Shadow Masking OFF



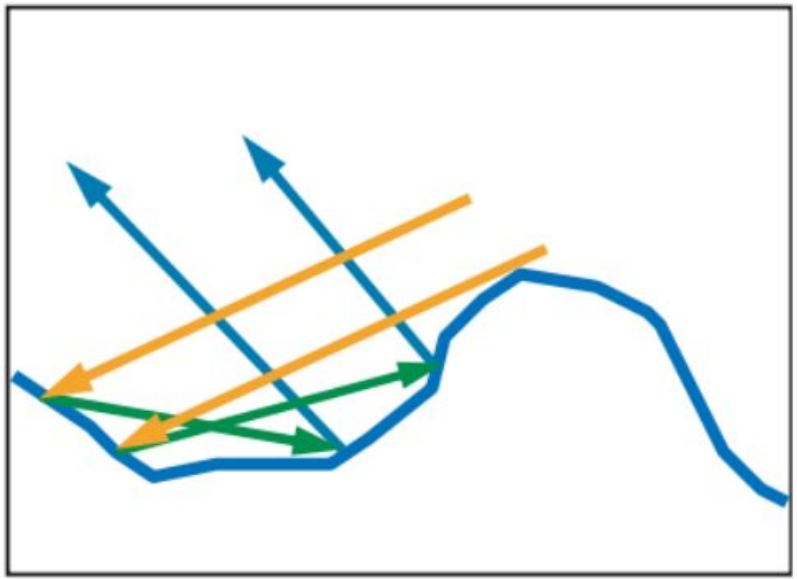
Trend Reed, Physically Based Shading and Image Based Lighting, http://www.trentreed.net/

Shadow Masking ON



Trend Reed, Physically Based Shading and Image Based Lighting, http://www.trentreed.net/

Multiple Bounces?



Puttying Specular BRDF together

Choose G and D depending on preference

$$f(\mathbf{l}, \mathbf{v}) = \frac{F(\mathbf{l}, \mathbf{h})G(\mathbf{l}, \mathbf{v}, \mathbf{h})D(\mathbf{h})}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})}$$

Diffuse Term

- Lambert Model
  - Equal scattering into all directions -> constant value
  - $\mathbf{c}_{\mathbf{diff}}$ reflection coefficient in linear RGB [0, 1]

$$f_{\mathrm{Lambert}}(\mathbf{l}, \mathbf{v}) = \frac{\mathbf{c}_{\mathrm{diff}}}{\pi}$$

- Oren-Nayar
  - Microfacet theory applied to diffuse scattering

Combining Diffuse & Specular

$$f(l,v) = f_{diff}(l,v) + f_{spec}(l,v)$$

- Actually only fraction of light not reflected by specular term can account for diffuse scattering
- Added Fresnel:

$$f_d = \frac{baseColor}{\pi}(1 - F(\theta_l))(1 - F(\theta_d))$$

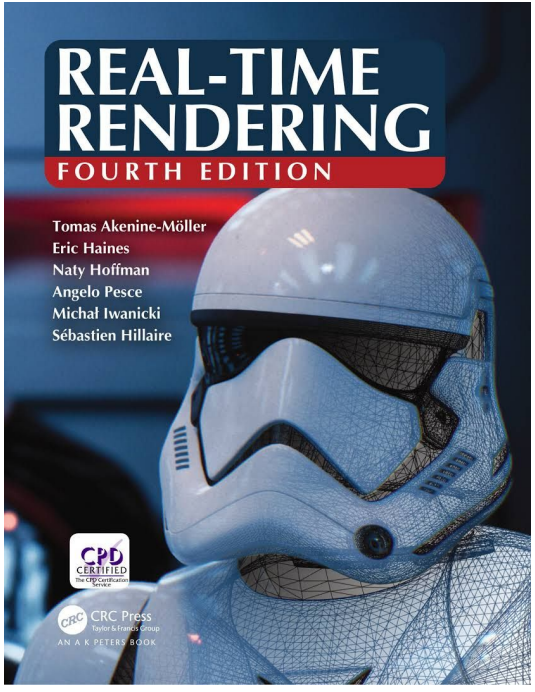
- Disney solution:

$$f_d = \frac{baseColor}{\pi}\left(1 + (F_{D90} - 1)(1 - \cos\theta_l)^5\right)\left(1 + (F_{D90} - 1)(1 - \cos\theta_v)^5\right)$$

$$\text{where} \quad F_{D90} = 0.5 + 2\cos\theta_d^2\, roughness$$

Resources

- Real-time Rendering, 4rd Edition, Tomas Akenine-Moller
- Physics and Math of Shading, Naty Hoffman, in SIGGRAPH Course Physically Based Shading Models for Film and Game Production 2010
  http://renderwonk.com/publications/s2010-shading-course/



SIGGRAPH Shading Courses 2010 - 2020!

Further Reading

- GGX, Beckmann, Blinn-Phong: Microfacet Models for Refraction through Rough Surfaces,
  Bruce Walter, Stephen R. Marschner, Hongsong Li, Kenneth E. Torrance, EGSR 2007,
  http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.3036&rep=rep1&type=pdf

- Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs,
  Eric Heitz, SIGGRAPH Course PBS 2014, JCGT 2014,
  Slides: http://blog.selfshadow.com/publications/s2014-shading-course/heitz/s2014_pbs_masking_shadowing_slides.pdf
  Paper: http://jcgt.org/published/0003/02/03/paper.pdf

Material Workflows

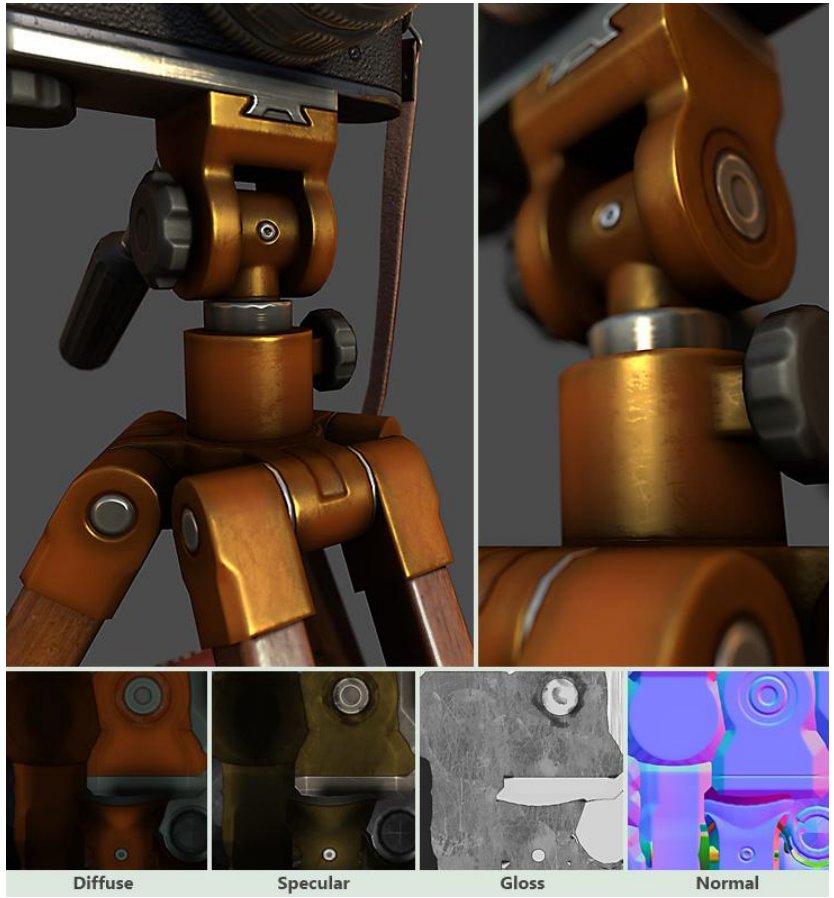
- **Specular**
  - Diffuse
  - Glossiness / Smoothness
  - Specular

- Metallic (originated from Disney)
  - Albedo / BaseColor
  - Metalness
  - Roughness
  - (Specular / Reflectance / custom IOR)

- Disney BRDF / OpenPBR
  - 10+ Parameters

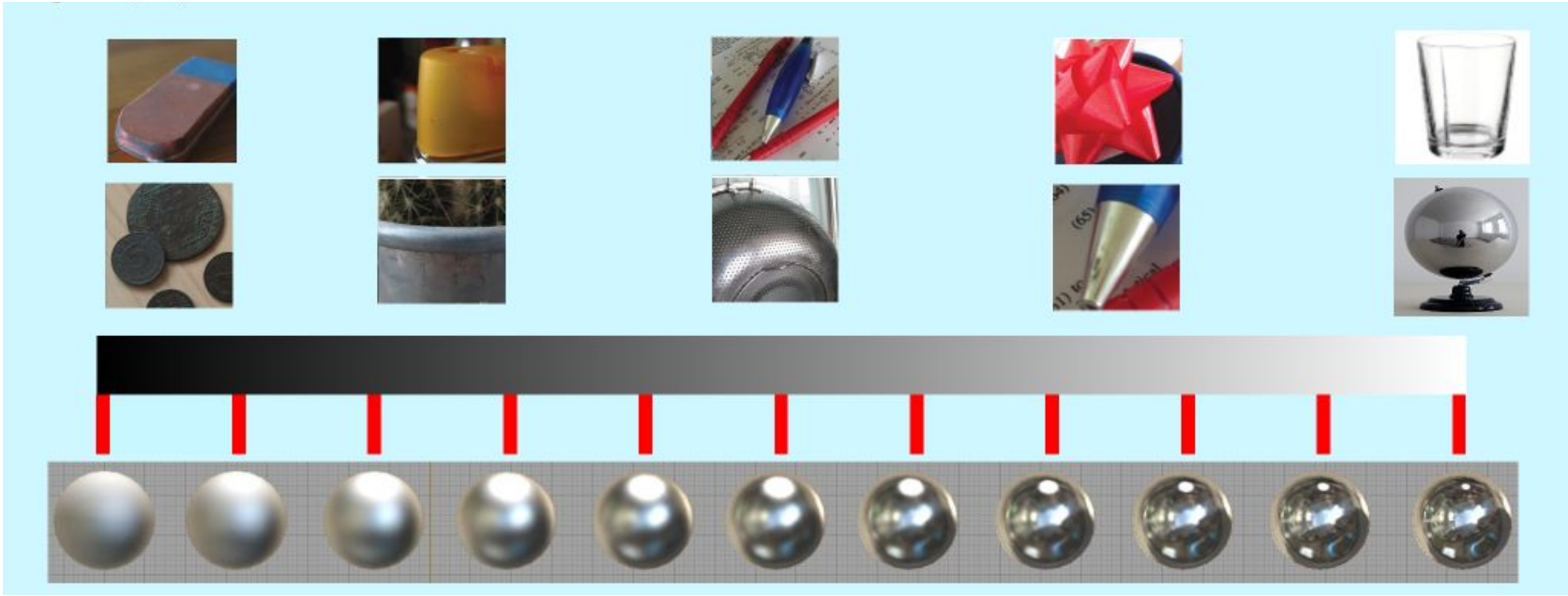
Specular Workflow

- Specular
  - Diffuse
  - Specular (Color / Strength)
  - Glossiness / Smoothness



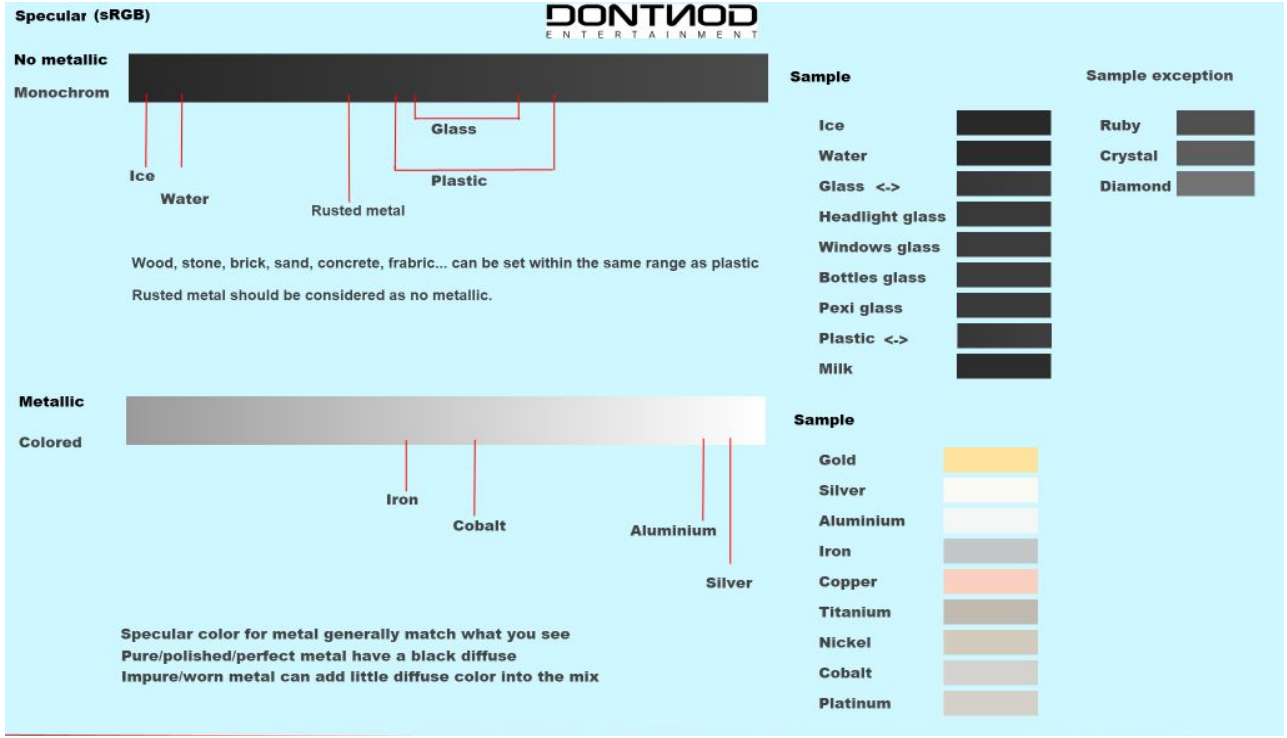
https://www.marmoset.co/posts/complex-material-setup/

Glossiness: Perceptual Linear Workspace



https://seblagarde.wordpress.com/2012/04/30/dontnod-specular-and-glossiness-chart/

Specular



https://seblagarde.wordpress.com/2012/04/30/dontnod-specular-and-glossiness-chart/

Specular vs Metalness

- Specular
  - Diffuse kd
  - Specular ks
  - Glossiness / Smoothness

- **Metallic**
  - Albedo / BaseColor
  - Metalness
  - Roughness (Inverted Glossiness)
  - Optionally: Reflectance / Custom IOR

- Gives full control to Artists
  - necessary?

kd = baseColor * (1.0 - metalness) * DiffuseShadow

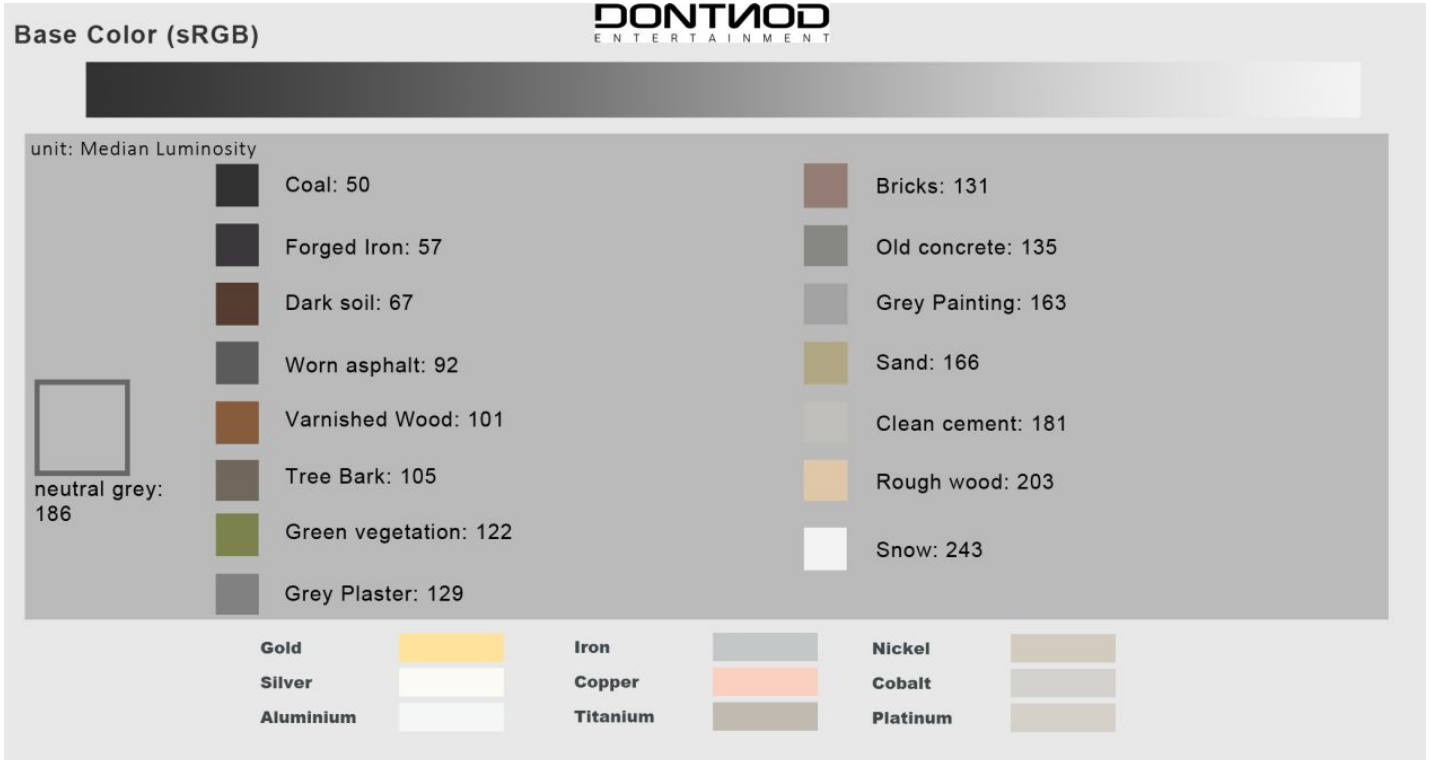
r0 = lerp(0.04, baseColor, metalness)
ks = F(r0, h) * GGX

Metalness Workflow

- Metallic
  - Albedo / BaseColor
  - Roughness
  - Metalness
  - Optionaly: Reflectance / custom IOR

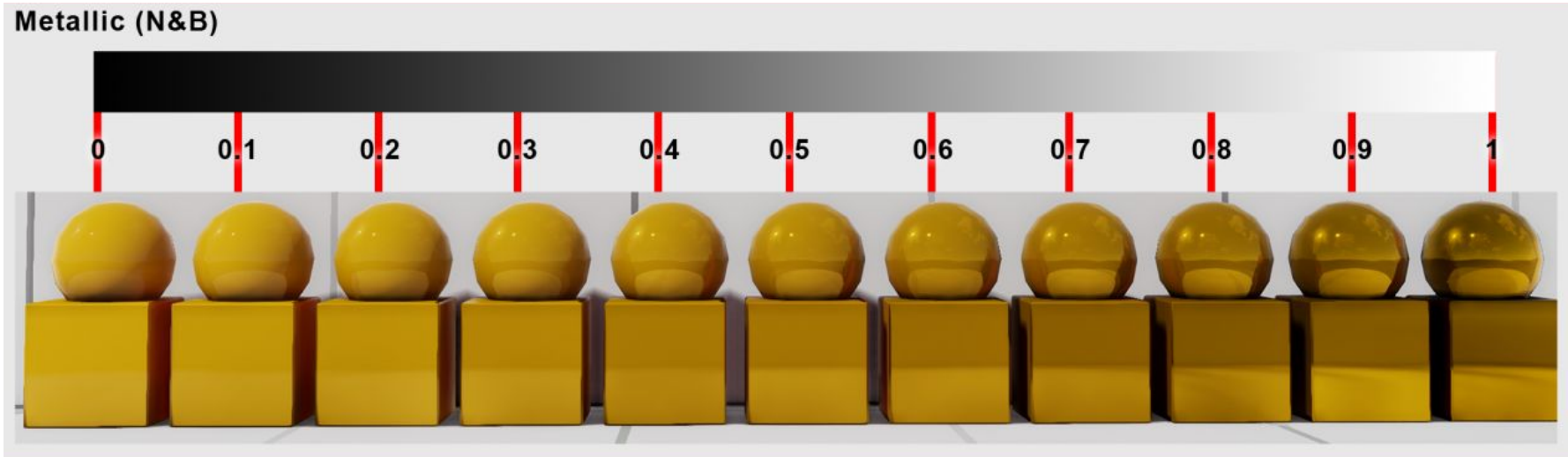
https://www.marmoset.co/posts/physically-based-rendering-and-you-can-too/

Base Color



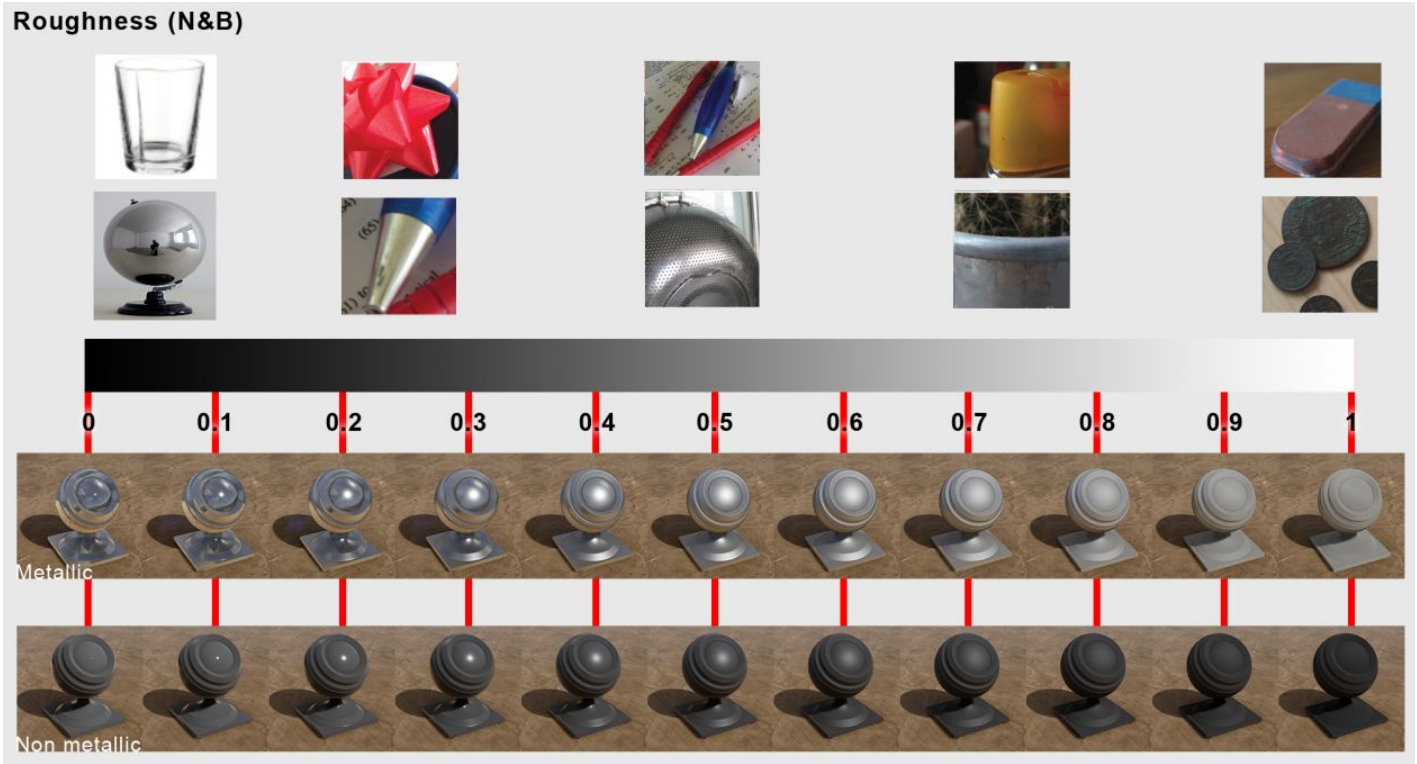
https://seblagarde.wordpress.com/2014/04/14/dontnod-physically-based-rendering-chart-for-unreal-engine-4/

Metalness



https://seblagarde.wordpress.com/2014/04/14/dontnod-physically-based-rendering-chart-for-unreal-engine-4/

Roughness: Perceptual Linear Workspace



https://seblagarde.wordpress.com/2014/04/14/dontnod-physically-based-rendering-chart-for-unreal-engine-4/

Material Workflow Conclusion

- Few parameters - easy to use
- Physically based workflow - consistency
- Specular & Metalness - support both?
- Need more? - Have a look at the Disney BRDF and OpenPBR
  - http://blog.selfshadow.com/publications/s2012-shading-course/burley/s2012_pbs_disney_brdf_slides_v2.pdf
  - https://github.com/AcademySoftwareFoundation/OpenPBR
- Communicate how to use!

Resource Summary (1)

Sébastien Largade:

Feeding a physically based shading model
https://seblagarde.wordpress.com/2011/08/17/feeding-a-physical-based-lighting-mode/

Specular and Glossiness Chart (PBR with Blinn-Phong)
https://seblagarde.wordpress.com/2012/04/30/dontnod-specular-and-glossiness-chart/

Physically based rendering chart (metalness workflow)
https://seblagarde.wordpress.com/2014/04/14/dontnod-physically-based-rendering-chart-for-unreal-engine-4/

Resource Summary (2)

Unreal Engine 5 Manual
https://dev.epicgames.com/documentation/en-us/unreal-engine/physically-based-materials-in-unreal-engine

glTF
https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html#appendix-b-brdf-implementation

Allegorithmic Substance Designer: Practical guidelines for creating PBR texture
https://www.allegorithmic.com/system/files/software/download/build/PBR_volume_02_rev05.pdf

Marmoset Toolbag: Physically-based rendering, and you can too!
https://www.marmoset.co/posts/physically-based-rendering-and-you-can-too/

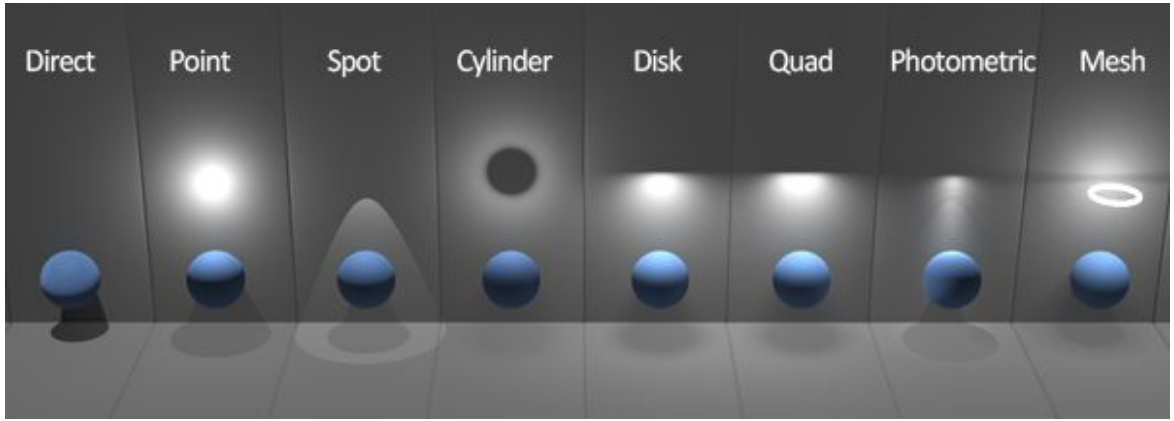
Evaluating the BRDF

- General Solution
- Point Lights
- Area Lights
- Image-based Lighting

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v})\, L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

Light Sources

- Point Light Sources
  - Point
  - Spot
  - Directional
  - Photometric
- Area Light Sources
  - Disk / Sphere
  - Tube
  - Rectangle
  - Polygon



Arnold Render

Point Light Definition

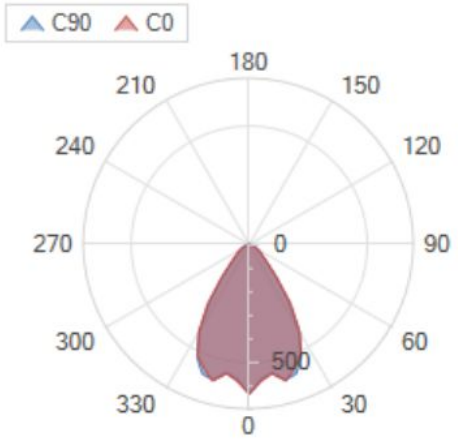
- Position / Orientation
- Color / Temperature in Kelvin
- Intensity
  - Luminous flux (lumen)
  - Luminous intensity (cd)

Color Temperature / Wikipedia

Point Light Definition

- Position / Orientation
- Color / Temperature in Kelvin
- Intensity
  - Luminous flux (lumen): total energy
  - Luminous intensity (cd): certain direction

1 lm = 1 cd · sr
Light bulb 100W: 1400 lumen / 4pi = 111cd

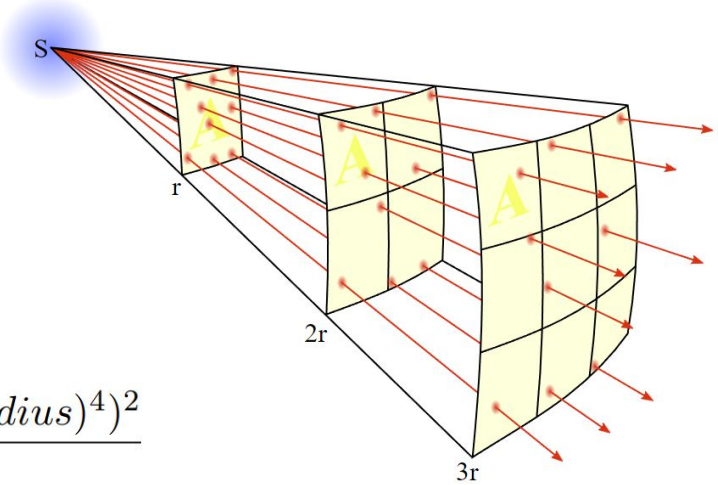


Light Photometry

Attenuation

- Inverse-square law: Intensity is inversely proportional to the square of the distance from the source

$$\frac{1}{distance^2}$$



- Need to limit range in practice
  - Solution in Unreal Engine 4:

$$\mathbf{falloff} = \frac{\text{saturate}(1 - (distance/lightRadius)^4)^2}{distance^2 + 1}$$

  - Frostbite: same with different bias in denominator

Wikipedia

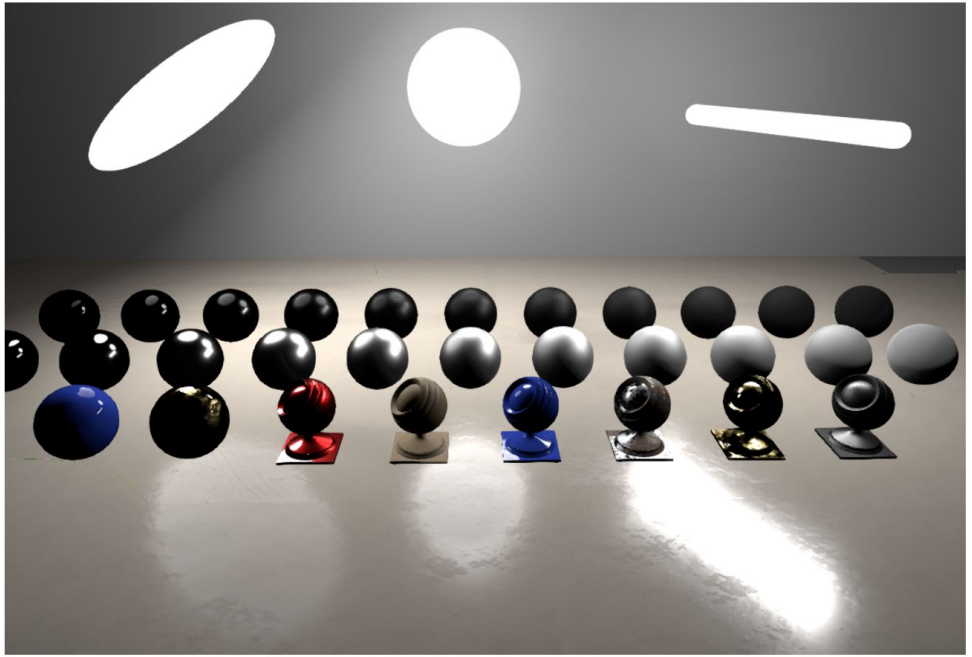
Point Light Evaluation

- Light emission into direction
- Attenuation / Geometric Term
- Visibility
  - E.g. Shadow Mapping
- BRDF

$$L_o(v) = L_e(l) * Attenuation * Visibility * f(l, v)(n \cdot l)$$

Area Light Source Definition

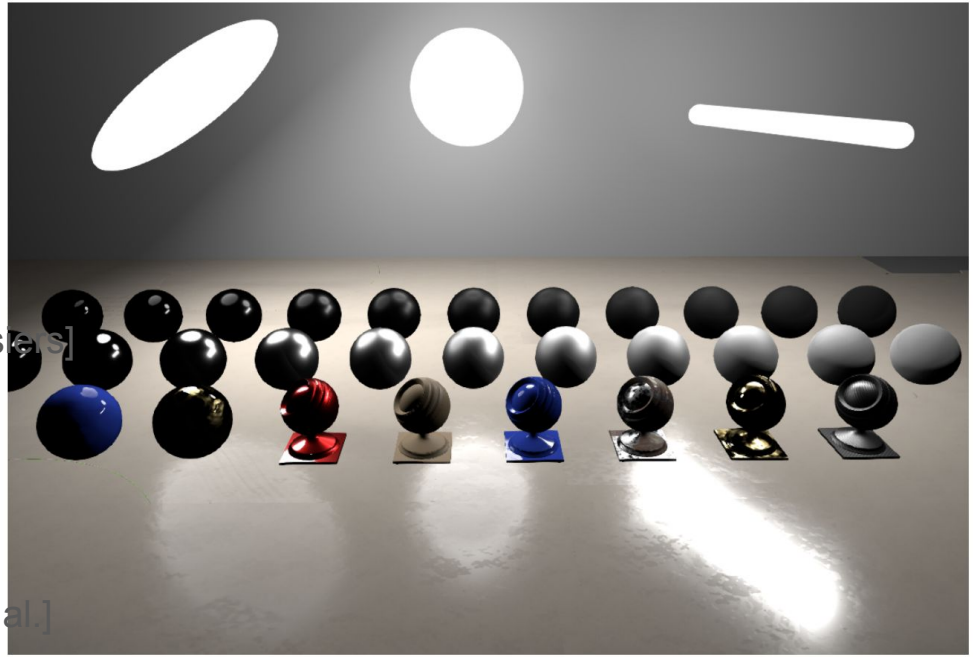
- Shape
  - Sphere / Disk
  - Tube / Linear
  - Rectangle
  - Polygon
- Color / Temperature in Kelvin
- Intensity / Luminance (cd/m²)



Frostbite

Area Light Evaluation

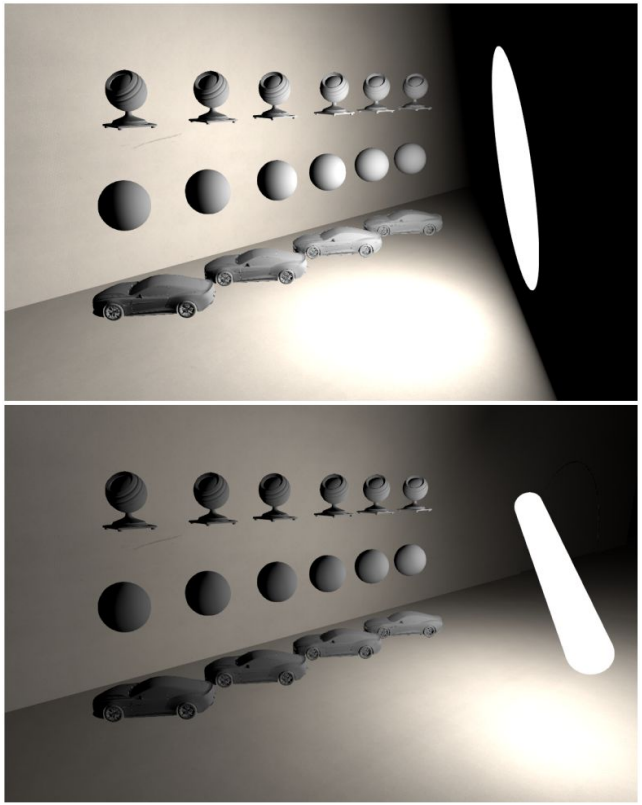
- **Need approximation**
  - Closed form: Le * f(v,l) * dot(l,n)
- Diffuse
  - Most Representative Point [Drobot]
  - Analytic Form Factor [Baum et al.]
  - Structured Sampling [Lagarde & Rousiers]
- Specular
  - Specular D modification [Mittring]
  - BRDF Widening [Karis]
  - Specular Area Overlap [Drobot]
  - Linear Transformed Cosines [Heitz et al.]
  - Analytic Solution [Lecocq et al.]



Frostbite

Area Light Evaluation

- Need approximation
  - Closed form: Le * f(v,l) * dot(l,n)
- **Diffuse**
  - Most Representative Point [Drobot]
  - Analytic Form Factor [Baum et al.]
  - Structured Sampling [Lagarde & Rousiers]
- Specular
  - Specular D modification [Mittring]
  - BRDF Widening [Karis]
  - Specular Area Overlap [Drobot]
  - Linear Transformed Cosines [Heitz et al.]
  - Analytic Solution [Lecocq et al.]



Frostbite

Area Light Evaluation

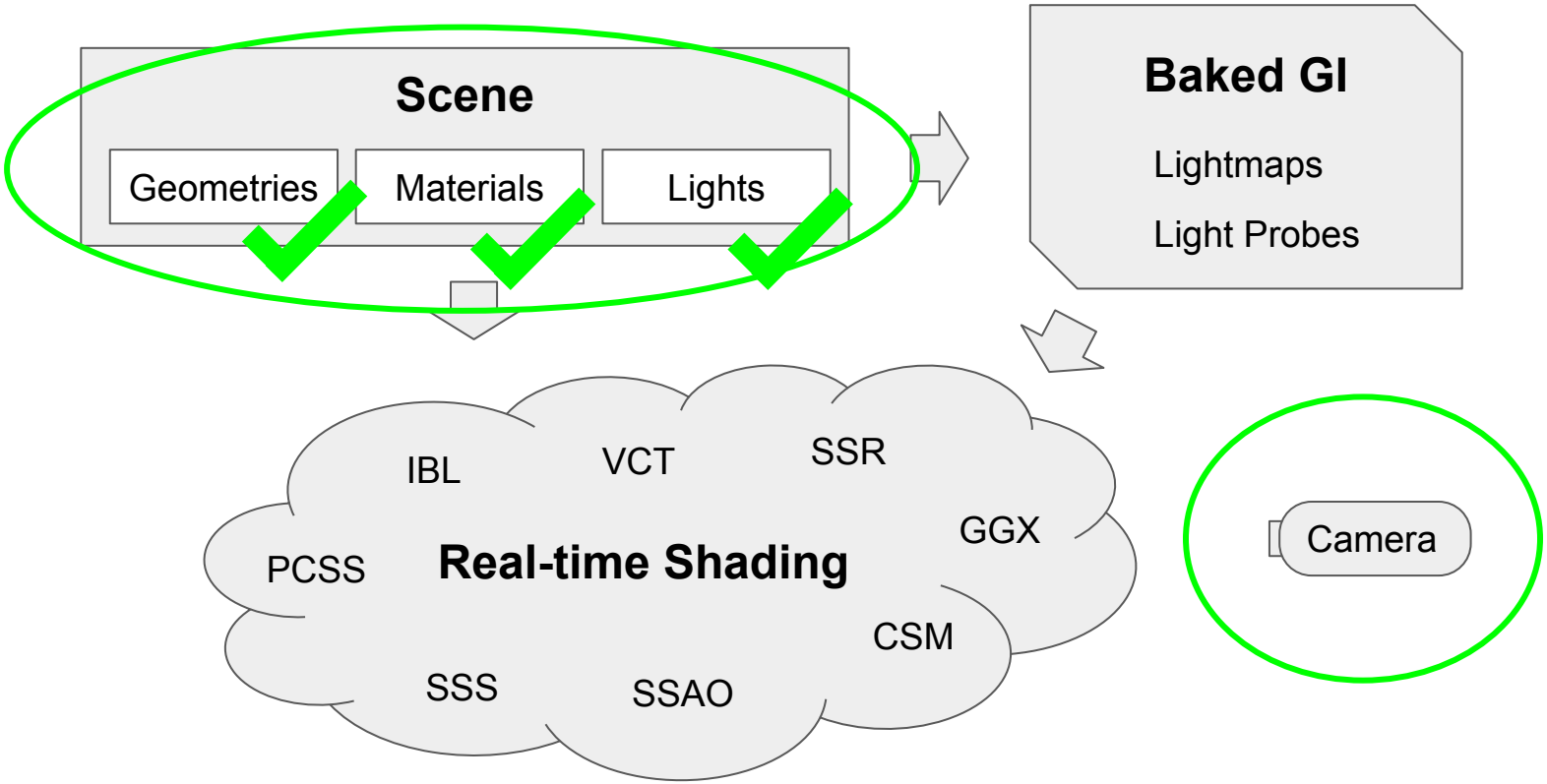
- Need approximation
  - Closed form: Le * f(v,l) * dot(l,n)
- Diffuse
  - Most Representative Point [Drobot]
  - Analytic Form Factor [Baum et al.]
  - Structured Sampling [Lagarde & Rousiers]
- **Specular**
  - Specular D modification [Mittring]
  - BRDF Widening [Karis]
  - Specular Area Overlap [Drobot]
  - Linear Transformed Cosines [Heitz et al.]
  - Analytic Solution [Lecocq et al.]

Polygon Lights with Linearly Transformed Cosines, Heitz et al.
https://eheitzresearch.wordpress.com/415-2/

Resources

- Moving Frostbite to Physically Based Rendering 3.0, Sébastien Lagarde, Charles de Rousiers, SIGGRAPH 2014 Course: PBS in Theory and Praxis
  Materials: https://seblagarde.wordpress.com/2015/07/14/siggraph-2014-moving-frostbite-to-physically-based-rendering/
- Real Shading in Unreal Engine 4, Brian Karis,
  SIGGRAPH 2013 Course: PBS in Theory and Praxis
  Slides: http://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013_pbs_epic_slides.pdf
  Notes: http://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013_pbs_epic_notes_v2.pdf
- Physically Based Area Lights, Michal Drobot, GPU Pro 5
- Real-Time Polygonal-Light Shading with Linearly Transformed Cosines, Heitz et al., SIGGRAPH 2016
  https://eheitzresearch.wordpress.com/415-2/
- Accurate Analytic Approximations for Real-time Specular Area Lighting, Lecocq et al. I3D 2016
  http://jmarvie.free.fr/Publis/2016_I3D/i3d_2016_AccurateAnalyticApproximationsForReal-TimeSpecularAreaLighting.pdf

Rendering System



Gamma

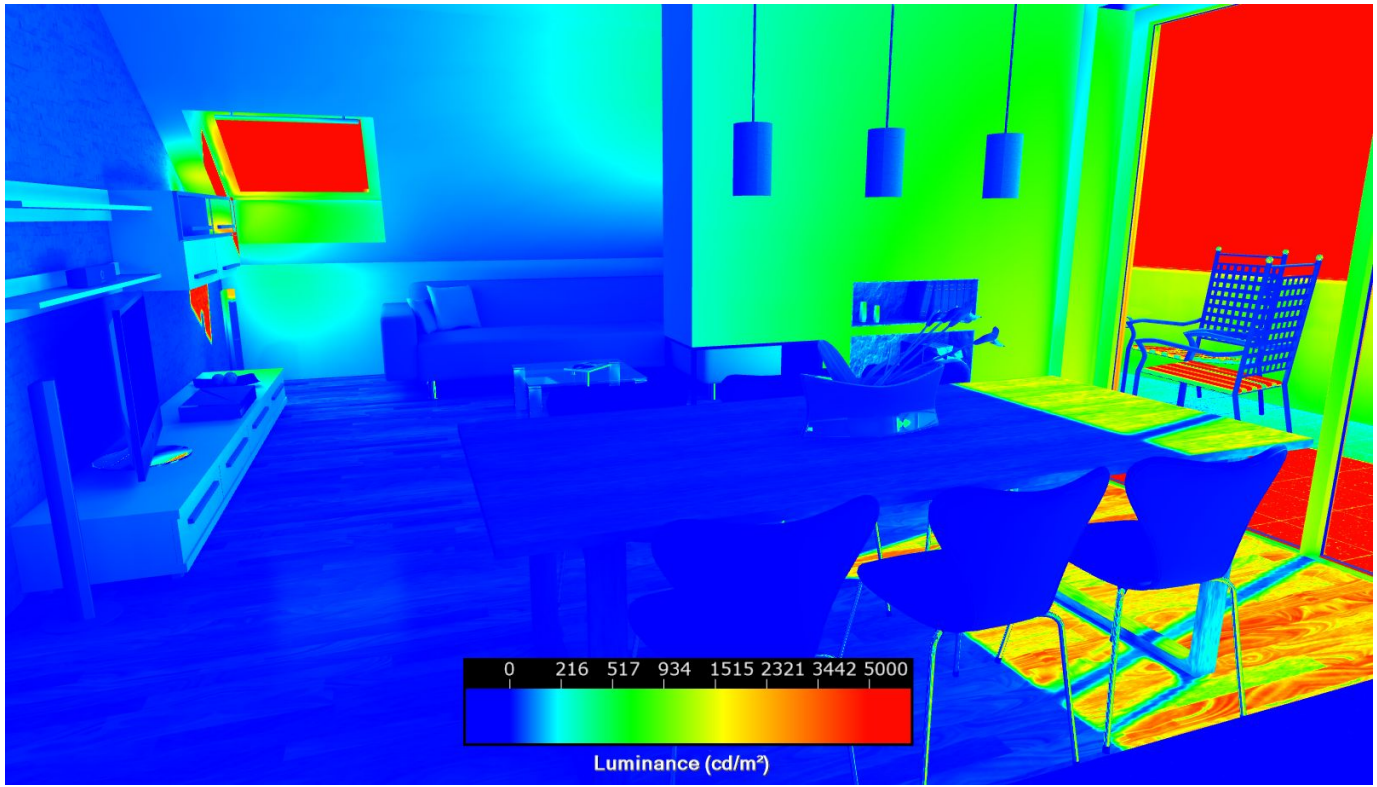
- Calculate everything in a linear color space!
  - Textures/Output framebuffer usually sRGB with gamma 2.2
  - Colors picked in UIs are usually also gamma 2.2
  - Make sure c_{diff} and c_{spec} shader uniforms are linear as well
  - Use floating point textures for intermediate targets and stay in linear until output
- Use Hardware conversions
  - sRGB texture format for textures and output framebuffers:

    ```
    glTexImage2D(target, level, sRGB Format Here, w, h, 0, pf, pt, 0);
    ```
  - Enable sRGB framebuffer write:

    ```
    glEnable(GL_FRAMEBUFFER_SRGB);
    ```

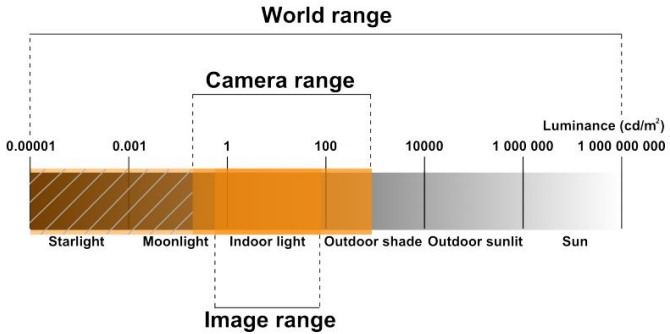
Dynamic Range

Dynamic Range



Tone Mapping

- Map luminance values to display device



- Have a look at how cameras work
  - Aperture, Shutter Time, Sensitivity -> Exposure (EV)
- Have a look at how human vision work

https://en.wikipedia.org/wiki/Exposure_value

https://placeholderart.wordpress.com/2014/11/21/implementing-a-physically-based-camera-manual-exposure/

Tone Mapping an Image

- Measure average luminance of image
  - Relative luminance of pixel given by:

$$L_w(x,y) = \overline{RGB}_{xy} \cdot \begin{pmatrix} 0.2126 \\ 0.7152 \\ 0.0722 \end{pmatrix}$$

  - Logarithmic average suitable for stable result:

$$\bar{L}_w = \exp\left(\frac{1}{N}\sum\left(\log\left(\delta + L_w(x,y)\right)\right)\right)$$

- Apply global tone mapping operator
  - Reinhard with user controlled key a:

$$L_{scaled} = a \cdot \frac{L_w}{\bar{L}_w} \qquad Color = \frac{L_{scaled} \cdot \left(1 + \frac{L_{scaled}}{L_{white}^2}\right)}{1 + L_{scaled}}$$

- Photographic tone reproduction for digital images

  Erik Reinhard, Michael Stark, Peter Shirley, James Ferwerda, SIGGRAPH 2002

  http://www.cs.utah.edu/~reinhard/cdrom/tonemap.pdf

Quick Implementation (1)

Use Compute Shaders for best performance!

1. Map HDR color buffer to temporary buffer (might just be 512^2)

```
// CIE XYZ D65 Luminance Weights
let LuminanceVector = V3d(0.2126, 0.7152, 0.0722)

let lumInit(v : Vertex) =
    fragment {
        let color = sceneInputSampler.Sample(v.tc).XYZ
        // calculate luminance
        let lum = Vec.dot color LuminanceVector
        // log output with clamp
        return V4d(clamp (log lum ) -10.0 20.0)
    }
```

2. Generate Mip-Levels

Quick Implementation (2)

Use Compute Shaders for best performance!

3. Tonemap HDR color buffer

```
let tonemap (v : Vertex) =
    fragment {
        let color = sceneInputSampler.Sample(v.tc).XYZ

        // get logarithmic average luminance
        let avgLum = exp (luminanceSampler.Read(V2i(0, 0), luminanceSampler.MipMapLevels - 1).X)

        let exposure = uniform.Key / avgLum
        let exposedColor = color * exposure

        // Reinhard tonemapper
        let iwls : float = uniform?InvWhiteLuminanceSquared
        return V4d(exposedColor * (1.0 + exposedColor * iwls) / (1.0 + exposedColor), 1.0)
    }
```

Or even skip 1 & 2 and use manual exposure value 😁

Additional Resources

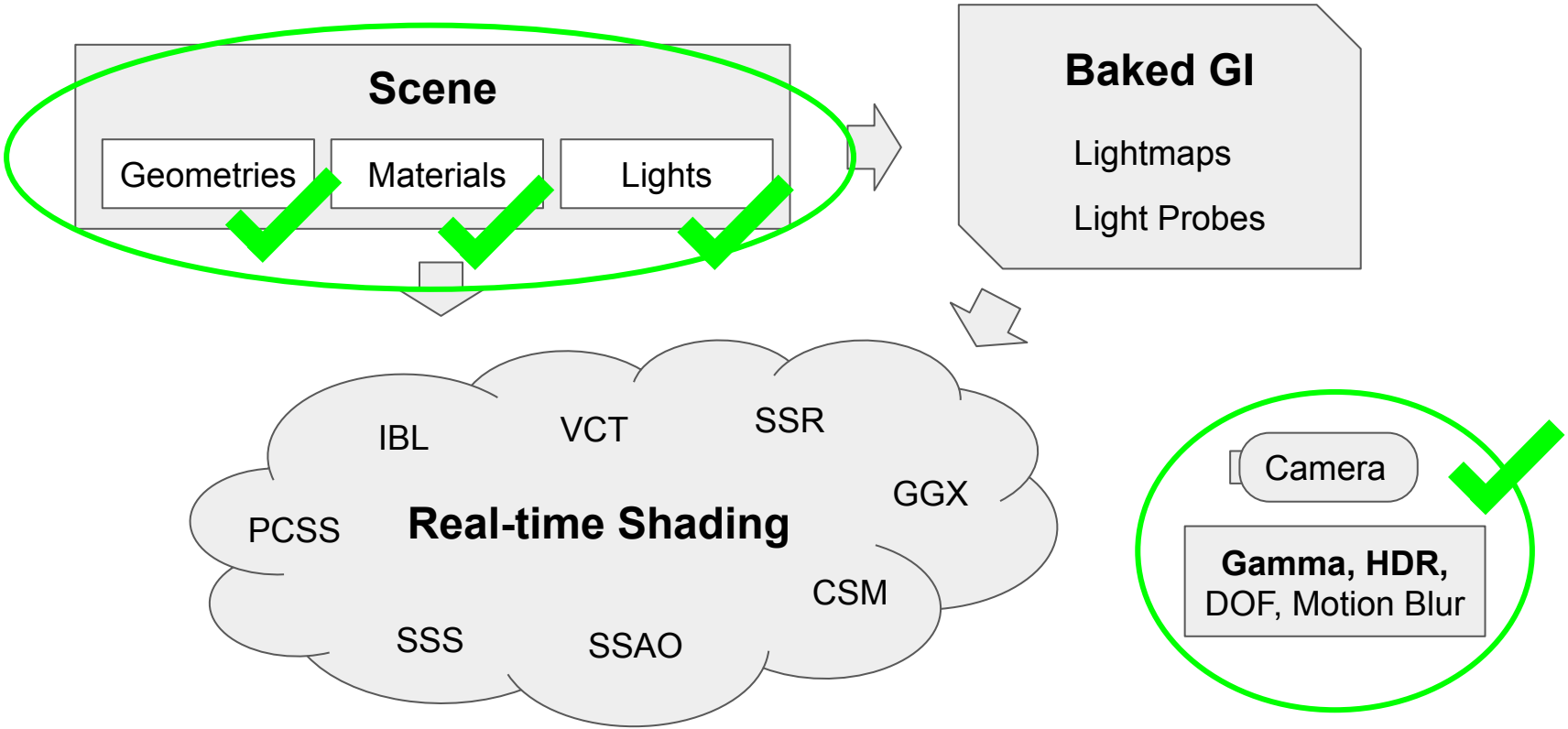
- A Closer Look At Tone Mapping, Matt Pettineo
  https://mynameismjp.wordpress.com/2010/04/30/a-closer-look-at-tone-mapping/

- Moving Frostbite to Physically Based Rendering 3.0, Sébastien Lagarde, Charles de Rousiers, SIGGRAPH 2014 Course: PBS in Theory and Praxis
  https://seblagarde.wordpress.com/2015/07/14/siggraph-2014-moving-frostbite-to-physically-based-rendering/

- Real-World Measurements for Call of Duty: Advanced Warfare, Danny Chan
  https://research.activision.com/publications/archives/real-world-measurements-for-call-of-dutyadvanced-warfare

- Photography-based Camera Exposure Control
  https://en.wikipedia.org/wiki/Exposure_value
  https://placeholderart.wordpress.com/2014/11/21/implementing-a-physically-based-camera-manual-exposure
  -> I would go for EV directly, but S, t, N also possible

Rendering System



Takeaways

- What is a BRDF
  - Diffuse & Specular
  - Generalized Cook-Torrance Model
- Material workflows
  - Specular vs Metalness
- Physical based light description
  - Point & Area
- Get physical units to the screen

Shader Modules Overview

Geometry	Light	Visibility	Filtering	BRDF	Post Effects
Displacement Tessellation Animation	Point Spot Area Directional Photometric Disk ...	Cubemap Perspective Paraboloid Cascaded Imperfect Stochastic ...	PCF Variance Poisson PCSS CSS …	Phong Ward GGX ...	DOF SSAO Scattering Tonemap

	Visibility	**Filtering**	**Area Light Approx**		
	+Image-based Lighting			**BRDF**	
	+Global Illumination			**BRDF**	

Next Lesson

- Towards Global Illumination in Real-time Rendering