Real Shading in Unreal Engine 4

Brian Karis (brian.karis@epicgames.com)

Goals

- More realistic image
- Material layering
  - Better workflow
  - Blended in shader
- Timely inspiration from Disney
  - Presented in this course last year

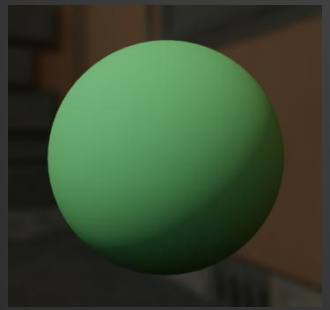
Overview

- Shading model
- Material model
- Lighting model

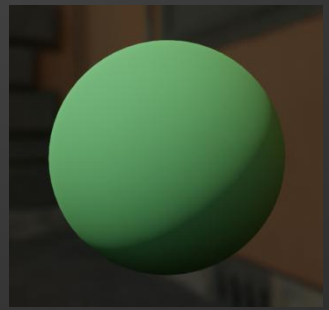
Shading Model

Diffuse BRDF

- Lambert
  - Saw little effect of more sophisticated models



Lambert



Burley

Specular BRDF

- Generalized microfacet model
  - Compared many options for each term
  - Use same input parameters

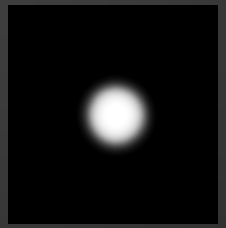
$$f(\mathrm{l},\mathrm{v}) = \frac{D(\mathrm{h})F(\mathrm{l},\mathrm{h})G(\mathrm{l},\mathrm{v},\mathrm{h})}{4(\mathrm{n}\cdot\mathrm{l})(\mathrm{n}\cdot\mathrm{v})}$$

Specular distribution

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

- Trowbridge-Reitz (GGX)
  - Fairly cheap
  - Longer tail looks much more natural

GGX

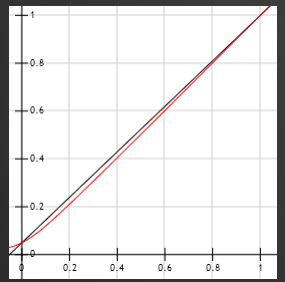


Blinn-Phong

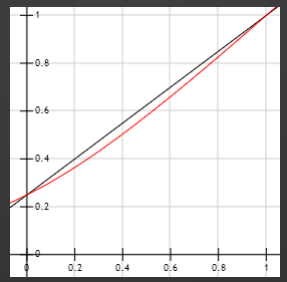
Geometric shadowing

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

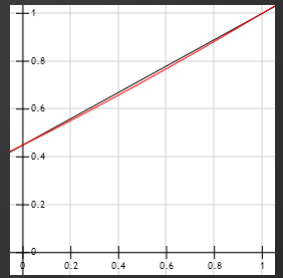
- Schlick
  - Matched to Smith
  - Cheaper, difference is minor
  - Uses Disney's roughness remapping*



$\alpha = 0.1$



$\alpha = 0.5$



$\alpha = 0.9$

$$\frac{n \cdot v}{G_{Schlick}(v)}$$

$$\frac{n \cdot v}{G_{Smith}(v)}$$

Fresnel

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

- Schlick
  - Approximate the power

Identical for all practical purposes

Image-based lighting : Problem

- Only use single sample per environment map
- Match importance-sampled reference

$$\int_H L_i(\mathrm{l}) f(\mathrm{l}, \mathrm{v}) \cos\theta_{\mathrm{l}} \, d\mathrm{l} \approx \frac{1}{N} \sum_{k=1}^{N} \frac{L_i(\mathrm{l}_k) f(\mathrm{l}_k, \mathrm{v}) \cos\theta_{\mathrm{l}_k}}{p(\mathrm{l}_k, \mathrm{v})}$$

Image-based lighting : Solution

- Same as Dimitar's: split the sum
- Pre-calculate both parts

$$\frac{1}{N}\sum_{k=1}^{N}\frac{L_i(\mathrm{l}_k)f(\mathrm{l}_k,\mathrm{v})\cos\theta_{\mathrm{l}_k}}{p(\mathrm{l}_k,v)} \approx \left(\frac{1}{N}\sum_{k=1}^{N}L_i(\mathrm{l}_k)\right)\left(\frac{1}{N}\sum_{k=1}^{N}\frac{f(\mathrm{l}_k,\mathrm{v})\cos\theta_{\mathrm{l}_k}}{p(\mathrm{l}_k,\mathrm{v})}\right)$$

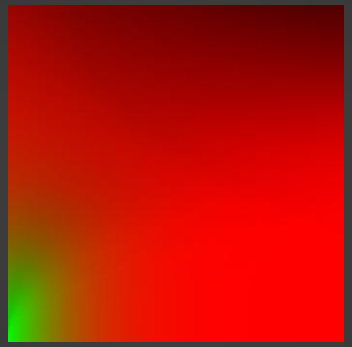
Pre-filtered environment map

- 1^{st} sum stored in cubemap mips
  - Pre-filter for specific roughness's
  - Fixed distribution, assume $\mathrm{n} = \mathrm{v}$
  - Loses stretched highlights

$$\frac{1}{N}\sum_{k=1}^{N} L_i(\mathrm{l}_k) \approx \mathrm{Cubemap.Sample}(\mathrm{r}, \mathrm{mip})$$

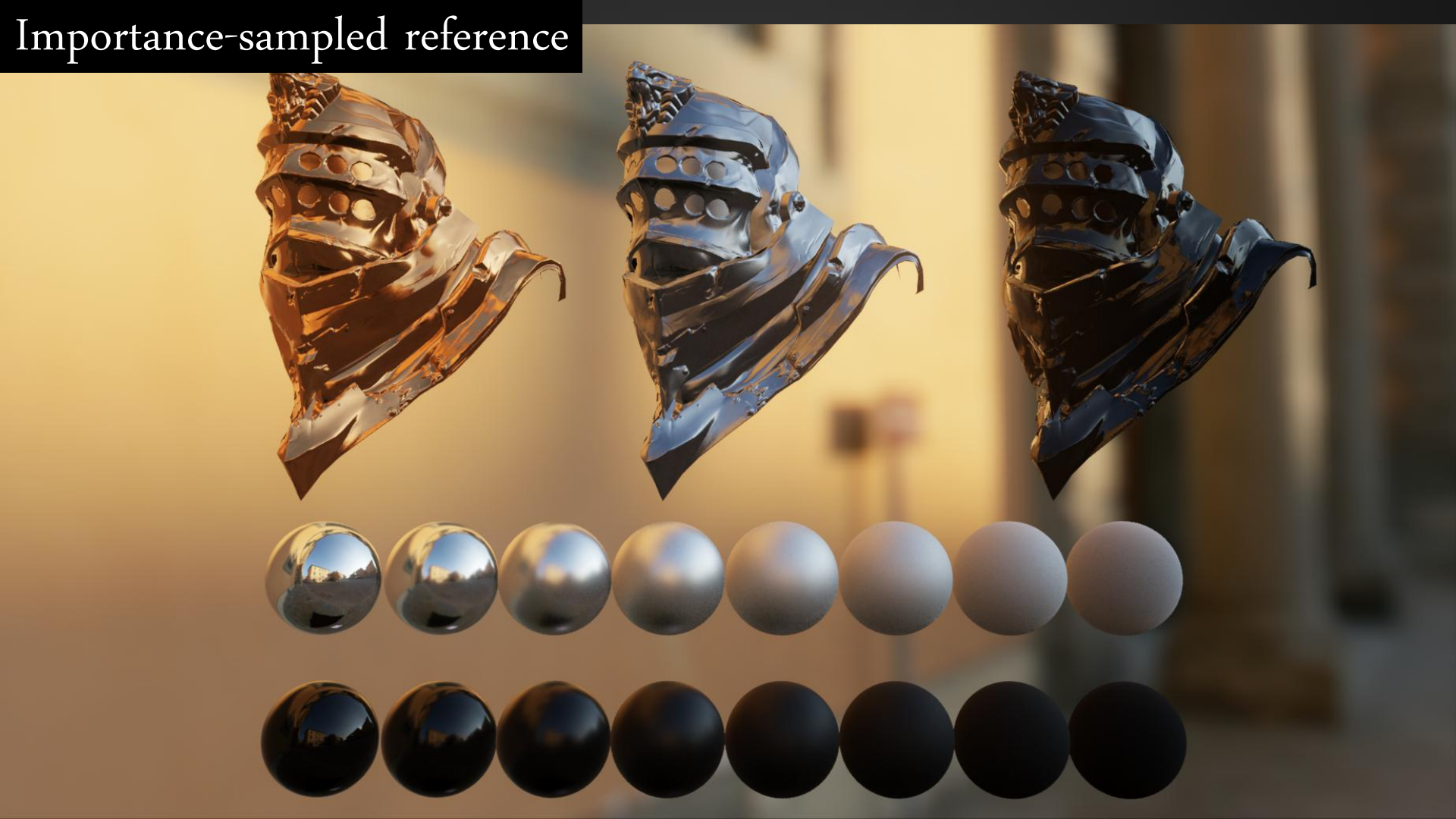
Environment BRDF

- 2^{nd} sum stored in 2D lookup texture (LUT)

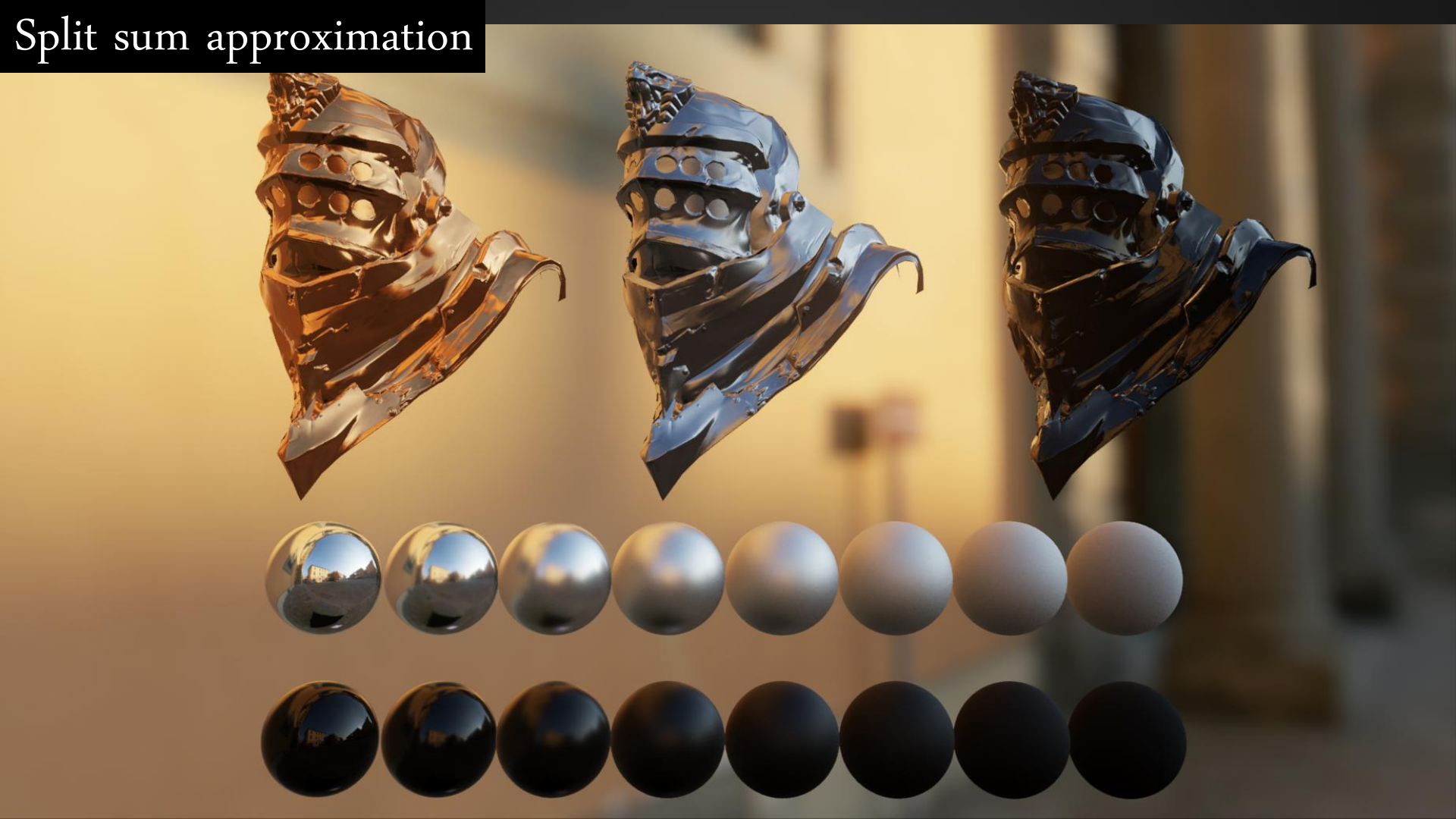


$$\frac{1}{N}\sum_{k=1}^{N}\frac{f(l_k, v)\cos\theta_{l_k}}{p(l_k, v)} = \text{LUT.r} * F_0 + \text{LUT.g}$$

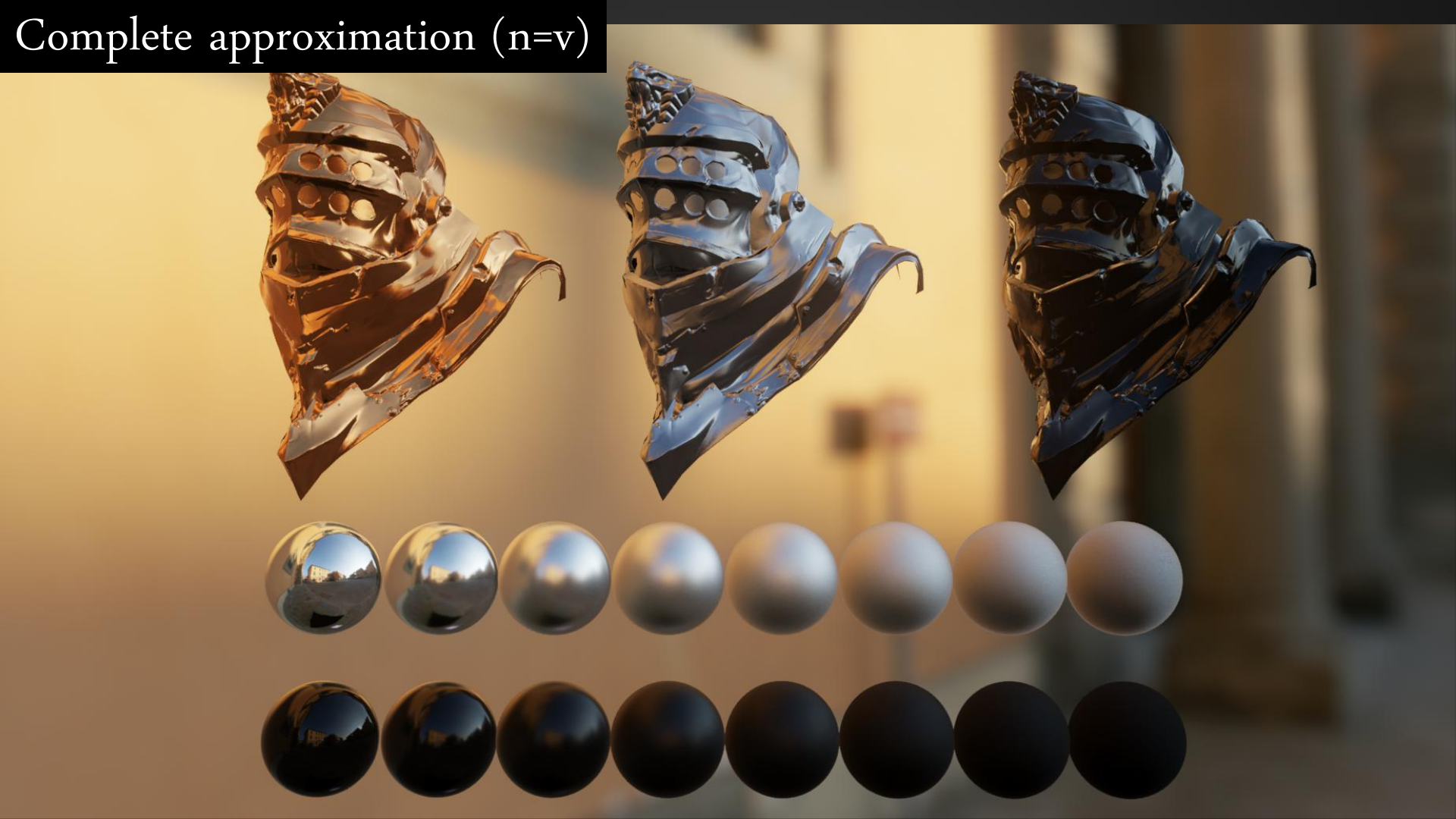
Importance-sampled reference



Split sum approximation

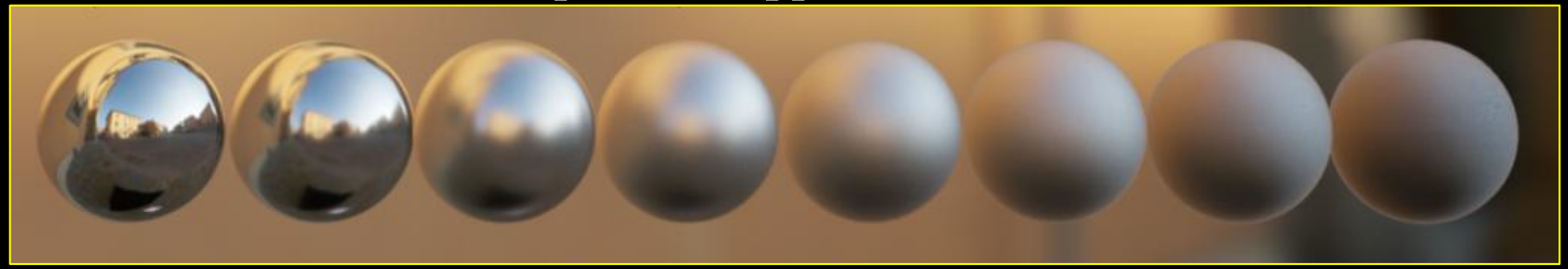


Complete approximation (n=v)



Importance-sampled reference

Split sum approximation



Complete approximation (n=v)

Material Model

Material model

- BaseColor
  - Single color
- Metallic
  - Less chance of error
- Roughness
  - Very clear in its meaning
- Cavity
  - Used for small scale shadowing

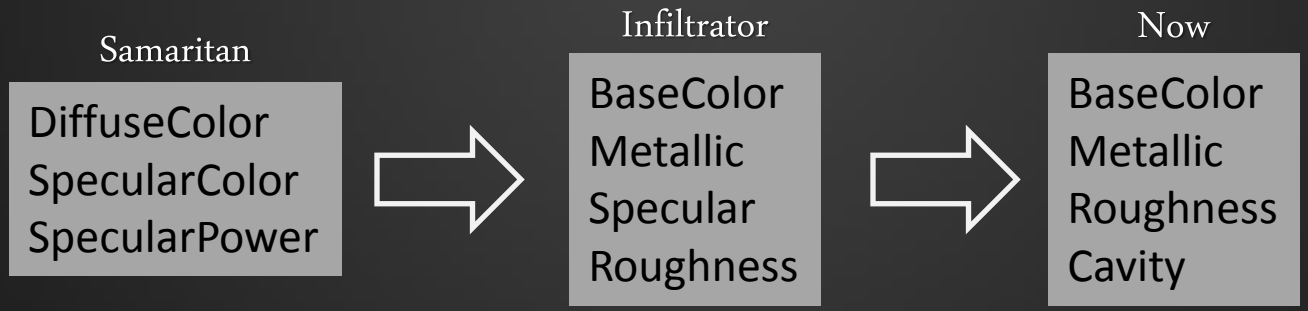
Metallic 0 to 1

Metal with roughness 0 to 1

Non-metal with roughness 0 to 1

Material model lessons

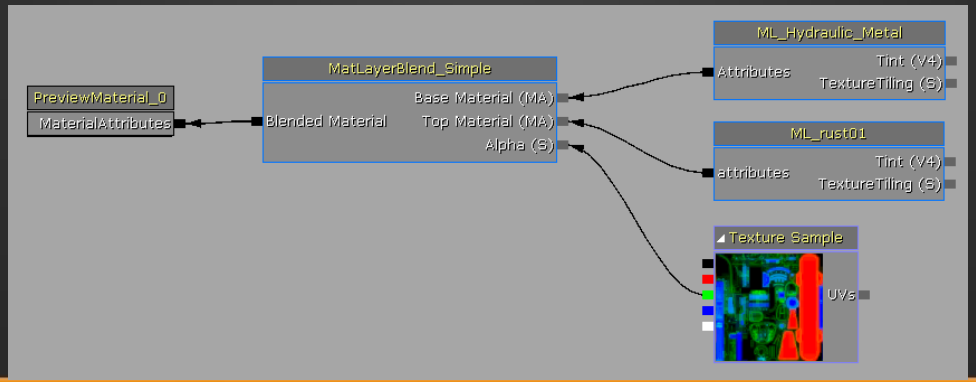
- Specular parameter is confusing
  - Not really needed
  - Replaced with Cavity



Material layering

Material layering tools

- Added layers to our node graph based material editor
  - Layers use existing material function feature
  - Added material attributes struct
- Layer workflow similar to previous texture workflow

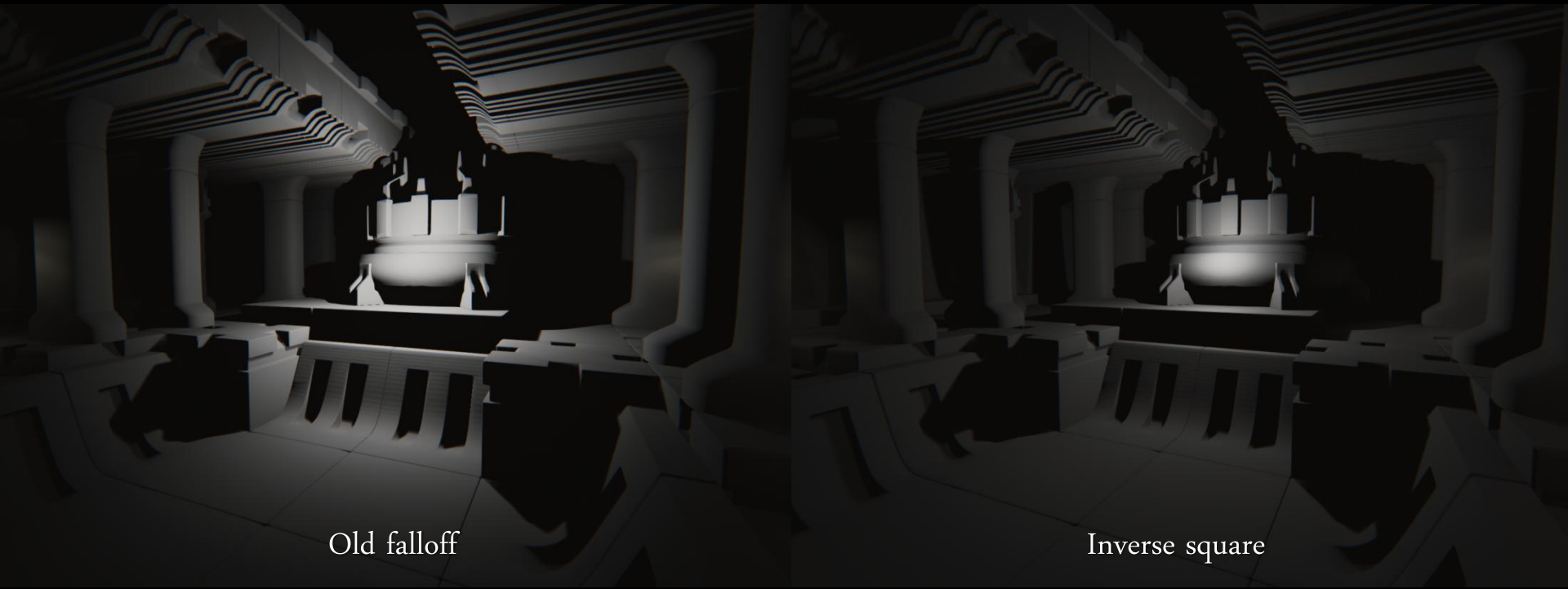


Material layering

Material layering

Lighting Model

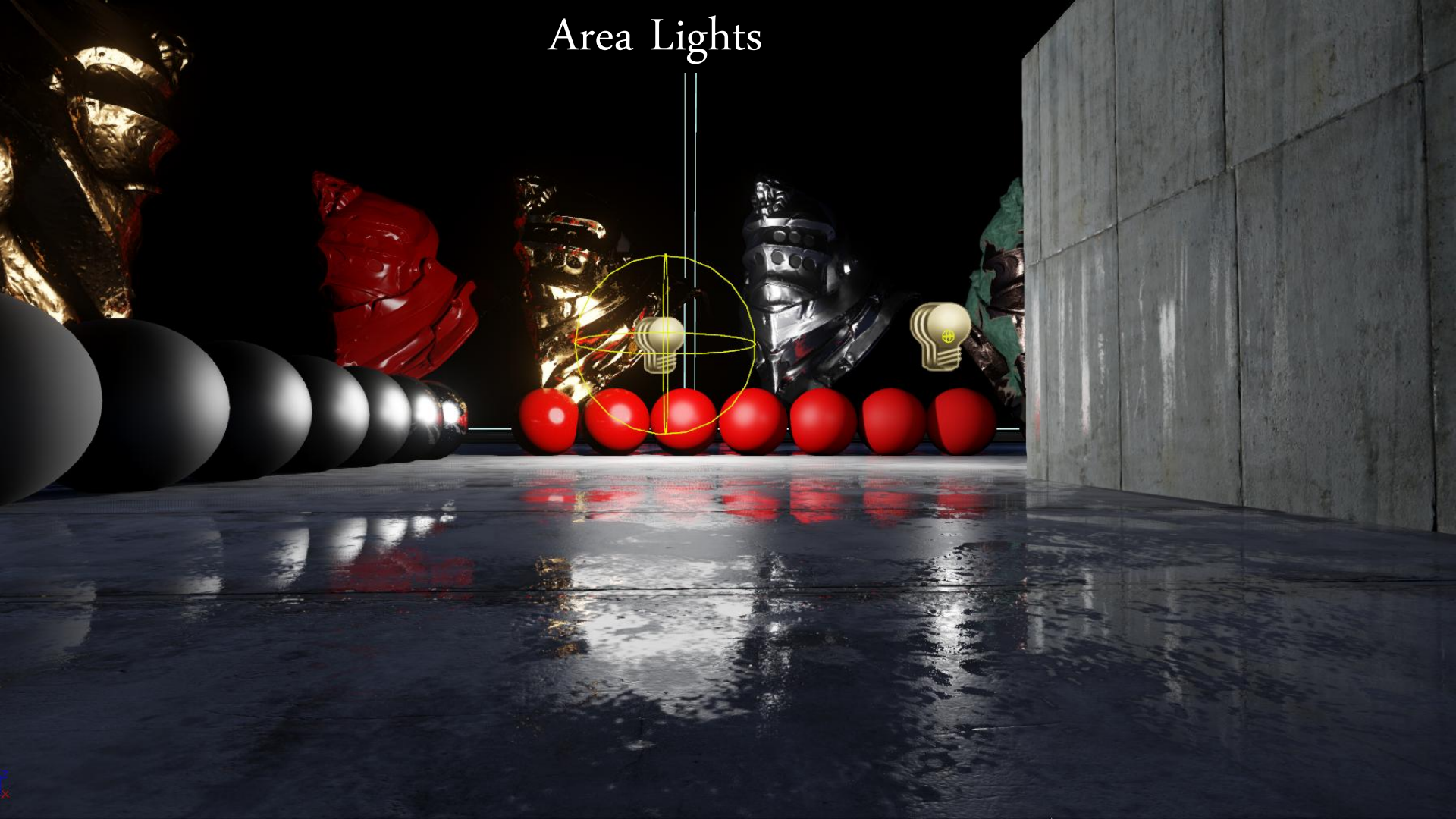
Inverse square falloff



Old falloff

Inverse square

Area Lights

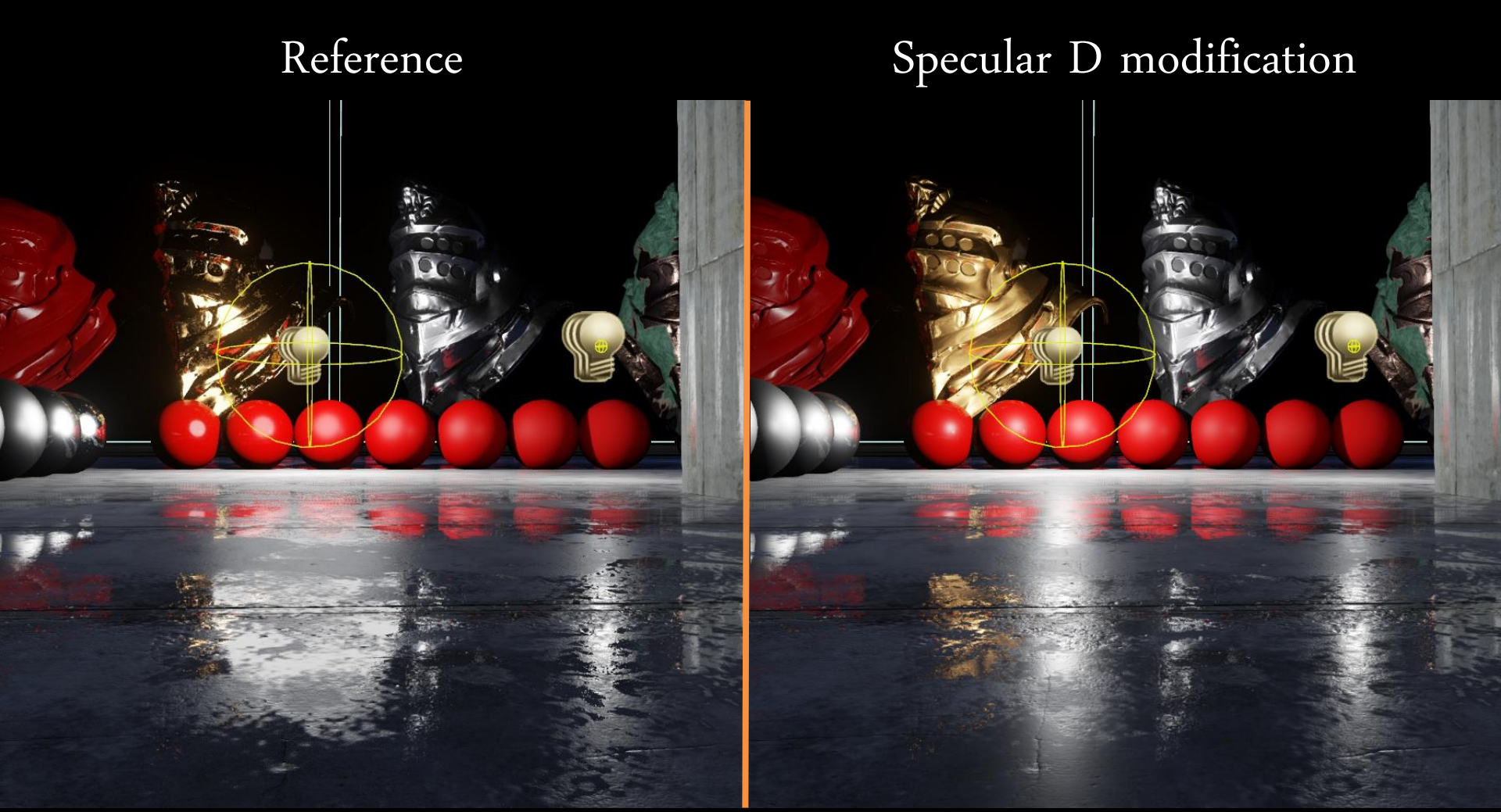


Area light requirements

- Consistent material appearance
  - Energy evaluated with diffuse BRDF and specular BRDF should match
- Approaches point light model as solid angle approaches zero
  - Don't want to lose any aspect of our shading model
- Fast enough to use everywhere
  - Otherwise artists will bias roughness

Specular D modification

- Widen specular distribution by light's solid angle
  - We presented this last year
- Problems
  - Glossy surfaces don't look glossy anymore

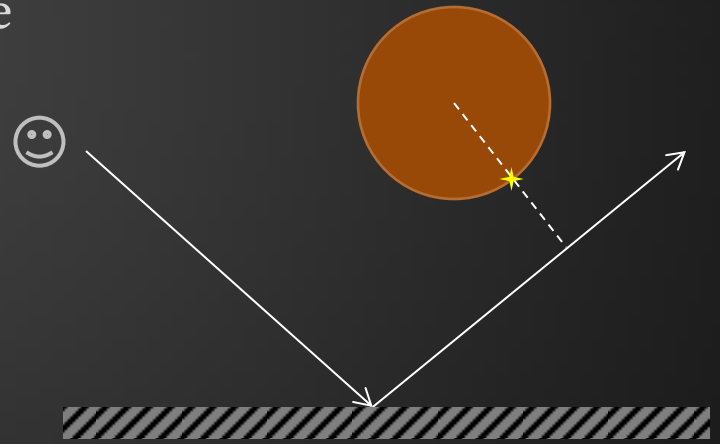

Reference
Specular D modification

Representative point

- Pick one representative point on light source shape
- Shading model can be used directly
- Point with largest contribution is a good choice
- Approximate using smallest angle to reflection ray

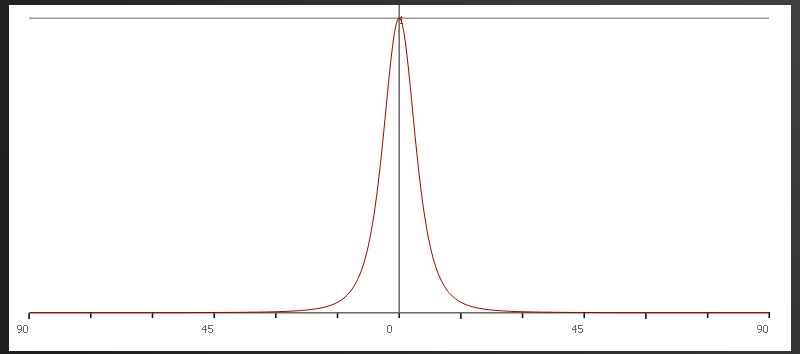
Sphere lights

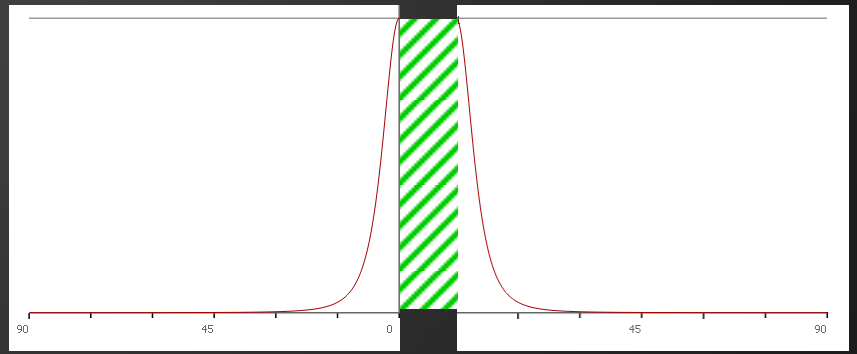
- Irradiance identical to point light
  - If sphere above horizon
- Closest point between ray and sphere
  - Approximates smallest angle

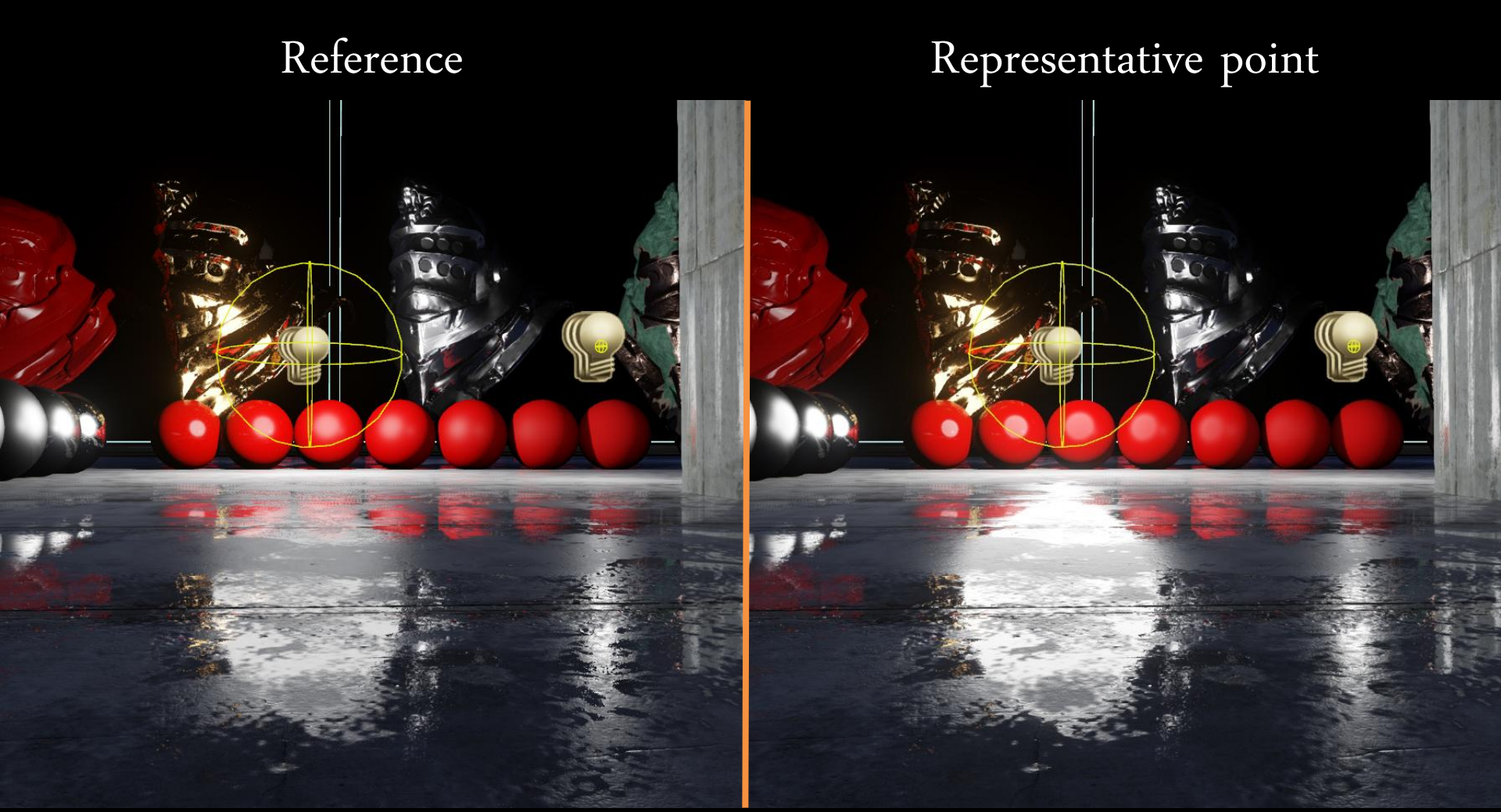


Sphere light energy conservation

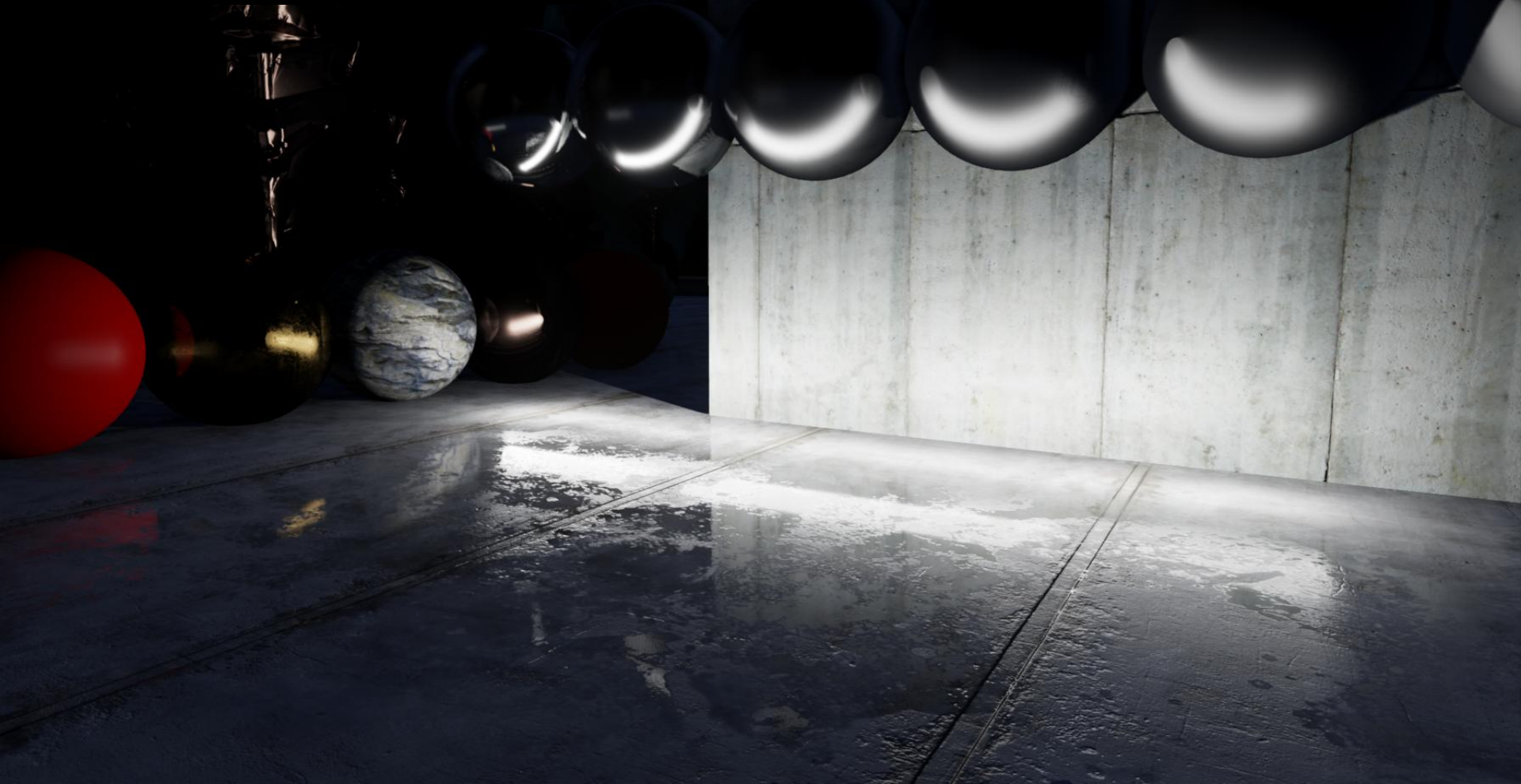
- Specular distribution has been widened by light's solid angle
  - We already have an approximation for this using "Specular D modification"
  - Only use normalization term
  - Divide out original normalization, multiply in new






Reference
Representative point

Representative point applied to Tube Lights



In the course notes

- Tons of extra stuff
  - Importance sampling code
  - Area light formulas
  - Lots of math ☺

Thanks

- Epic
  - Rendering team
  - All the artists making me look good
- Special thanks to Sébastien Lagarde
- Stephen Hill and Stephen McAuley for valuable input


EPIC
GAMES
We Are Hiring