Getting More Physical in CALL OF DUTY BLACK OPS II

Dimitar Lazarov

Lead Graphics Engineer, Treyarch

Black Ops: shading model

- Diffuse response
  - Direct: analytical lights
  - Indirect: lightmaps, light probes
  - Lambertian BRDF
- Specular response
  - Direct: analytical lights
  - Indirect: environment maps
  - Microfacet BRDF

Black Ops: Microfacet BRDF

- Based on Cook-Torrance:

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

$$D_{pl}(h) = \frac{\pi}{4} D(h)$$

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

$$D_{pl}(h)F(l,h)V(l,v,h)$$

* pl = point light

Black Ops: normal distribution function

- Blinn-Phong:

$$D_{pl}(h) = \frac{\alpha + 2}{8}(n \cdot h)^{\alpha}$$

$$\alpha = 8192^{g}$$

α: specular power

g: gloss

- Energy conserving
- Physically plausible stretchy highlights
- Cheaper replacement for Beckmann NDF (with parameter conversion)

Black Ops: reflectance function

- Schlick-Fresnel:

$$F(l,h) = rf_0 + (1 - rf_0)(1 - h \cdot l)^5$$

rf_0: base reflectance (specular color)

Black Ops: visibility function

- Schlick-Smith:

$$k = \frac{2}{\sqrt{\pi(\alpha + 2)}}$$

$$V(l, v, h) = \frac{1}{((n \cdot l)(1-k)+k)((n \cdot v)(1-k)+k)}$$

- Compared favorably to:
  - No visibility V(l, v, h*)* = 1
  - Cook-Torrance and Kelemen/Szirmay-Kalos (no gloss/roughness consideration)

Black Ops: environment map normalization

- Method to “fit” the environment map’s reflection to varying lighting conditions

Black Ops: normalization algorithm

Offline:

```
env_sh9 = capture_sh9(env_pos);
env_average_irradiance = env_sh9[0];
for_each (texel in environment map)
  texel /= env_average_irradiance;
```

Pixel Shader:

```
env_color = sample(env_map) * pixel_average_irradiance;
```

Black Ops: environment map pre-filtering

- Offline, CubeMapGen
  - Angular Gaussian filter
  - Edge fixup
- Pixel shader selects mip as a linear function of gloss:

```
texCUBElod(uv, float4(R, nMips - gloss * nMips));
```

Black Ops: environment map “Fresnel”

- More than just Fresnel, included shadowing-masking factor
- Early attempt at deriving an “Environment BRDF”

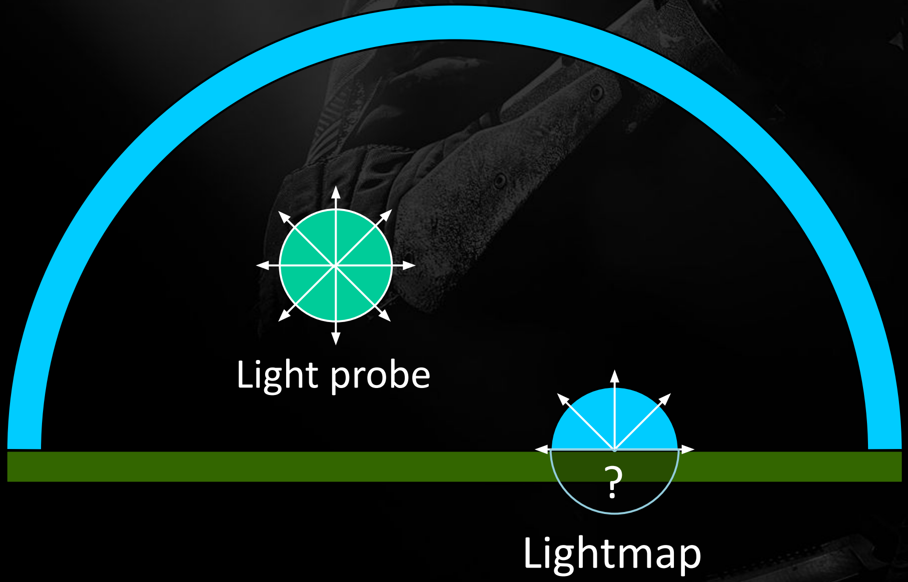
$$F(l,v) = rf_0 + (1 - rf_0)\frac{(1 - n \cdot v)^5}{4 - 3g}$$

Getting More Physical in Call of Duty: Black Ops II

- Direct Specular
  - Very happy with the look
  - Focused on performance improvements (details in the course notes)
- Indirect Specular
  - Various deficiencies in the Black Ops methods
  - The major focus of improvements

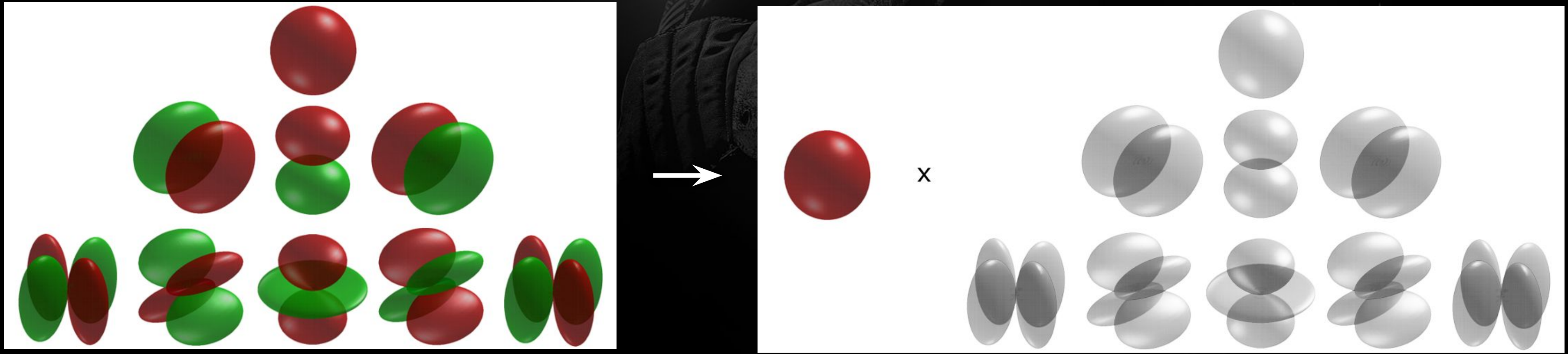
Environment map normalization: problem

- Average irradiance: poor choice for normalization



Environment map normalization: new idea

- Normalize with irradiance
  - Can't bake normalization offline
  - Pass environment map's directional irradiance to run-time (used tinted scalar 3rd-order Spherical Harmonics)



Improved normalization algorithm

Offline:

```
env_sh9 = capture_sh9(env_pos);
```

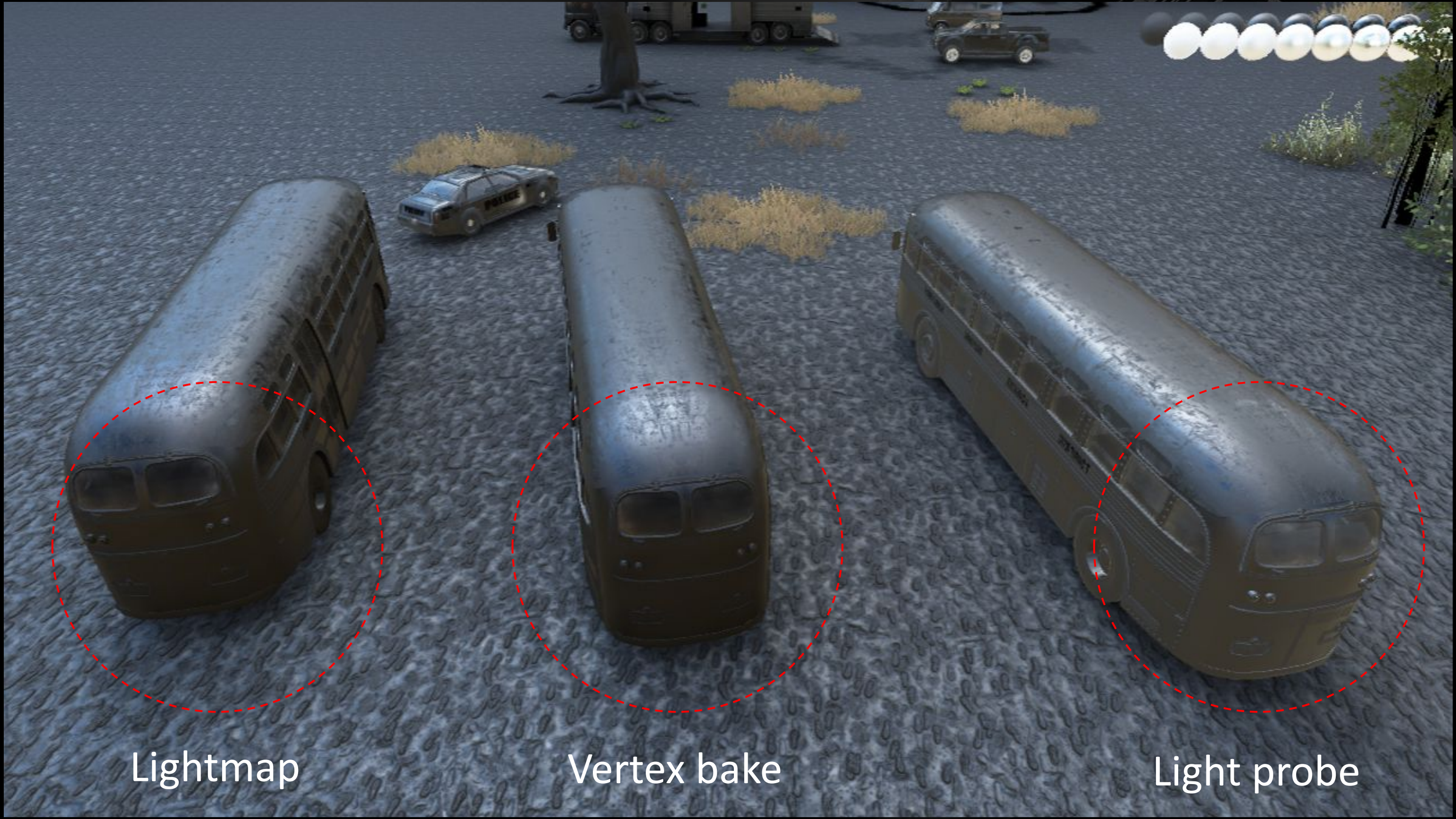
Vertex Shader:

```
env_irradiance = eval_sh(env_sh9, vertex_normal);
```

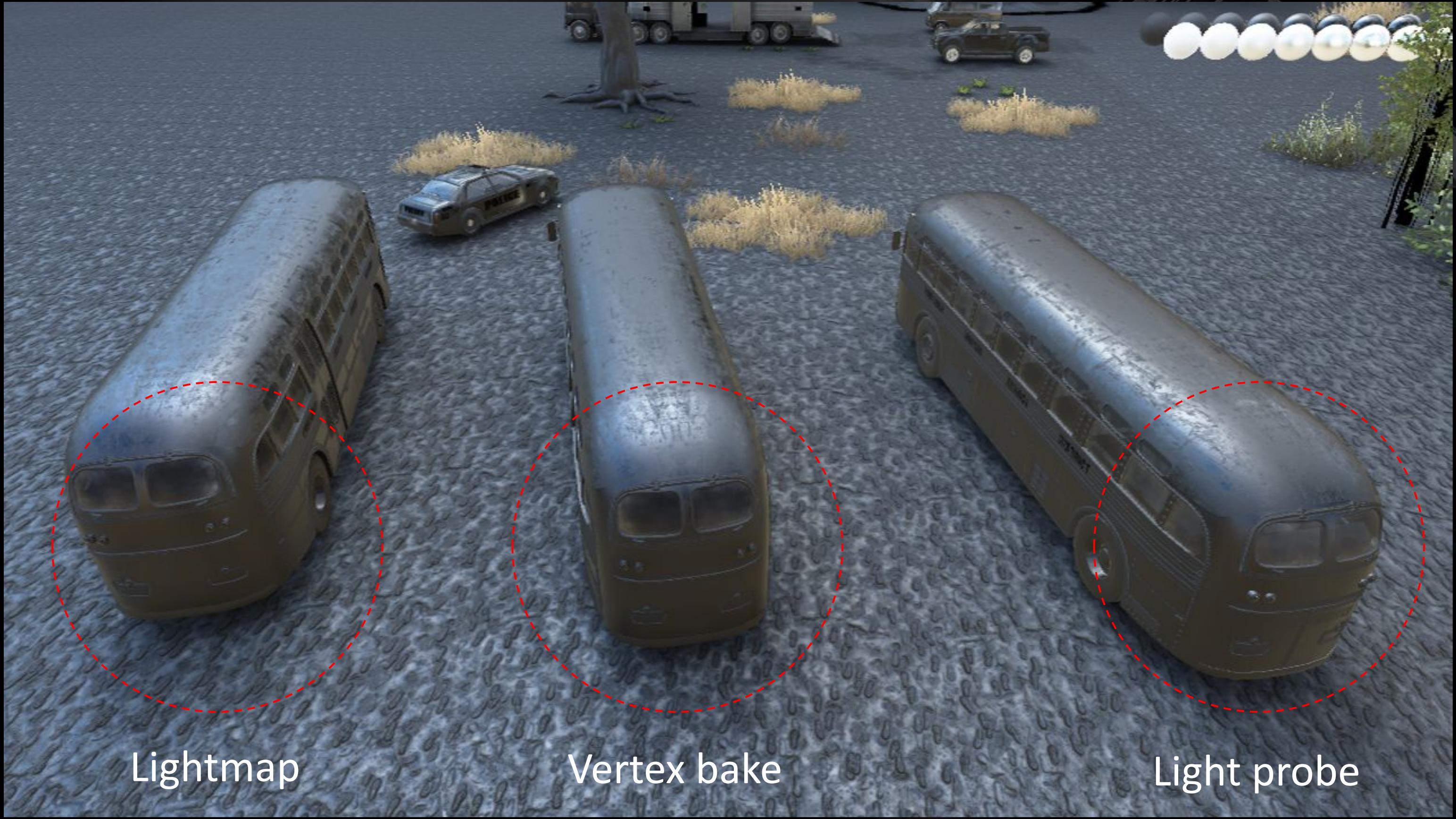
Pixel Shader:

```
env_color = sample(env_map)/env_irradiance * pixel_irradiance;
```

Environment map normalization: old method



Environment map normalization: new method

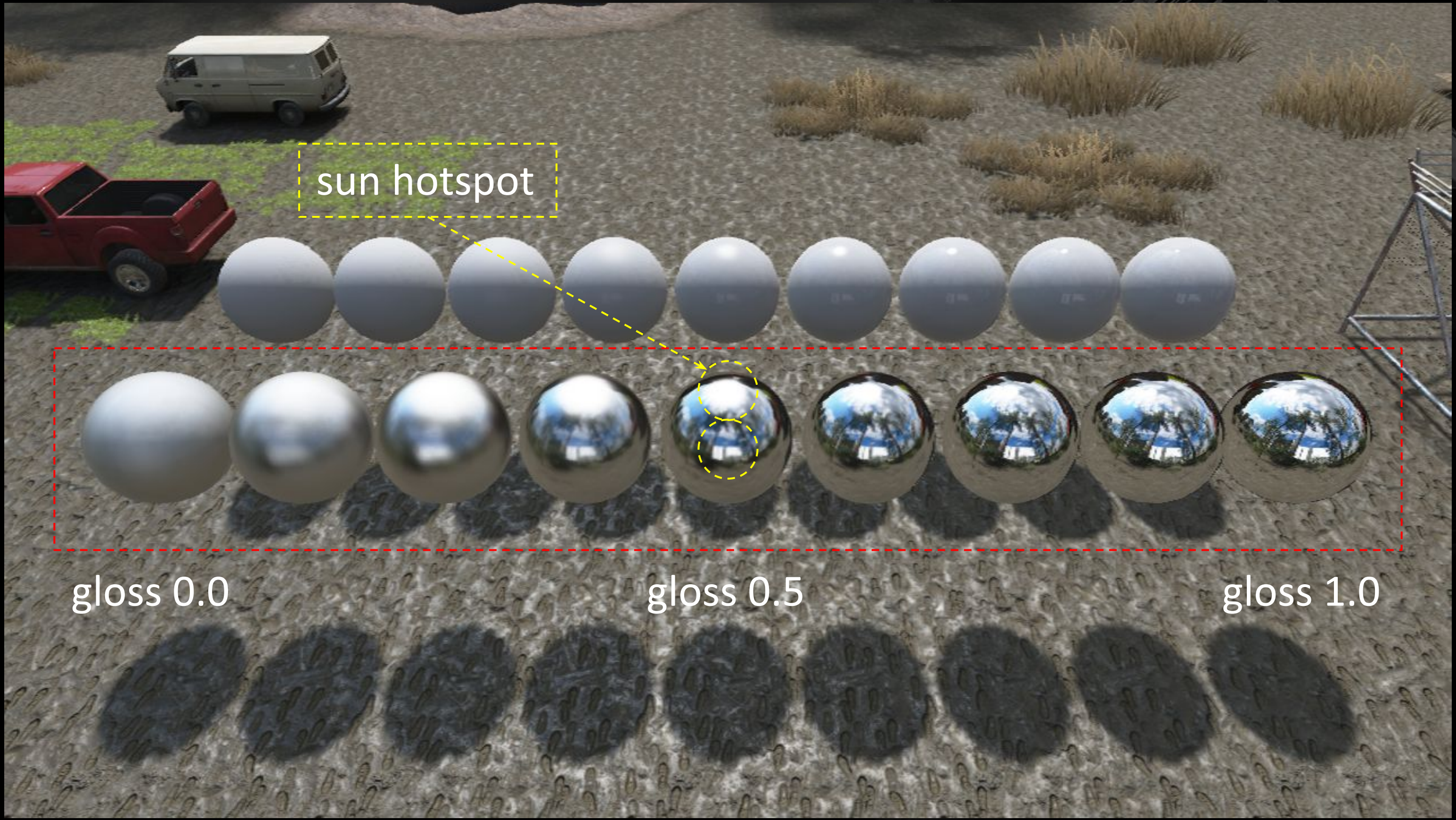


Improved environment map pre-filtering

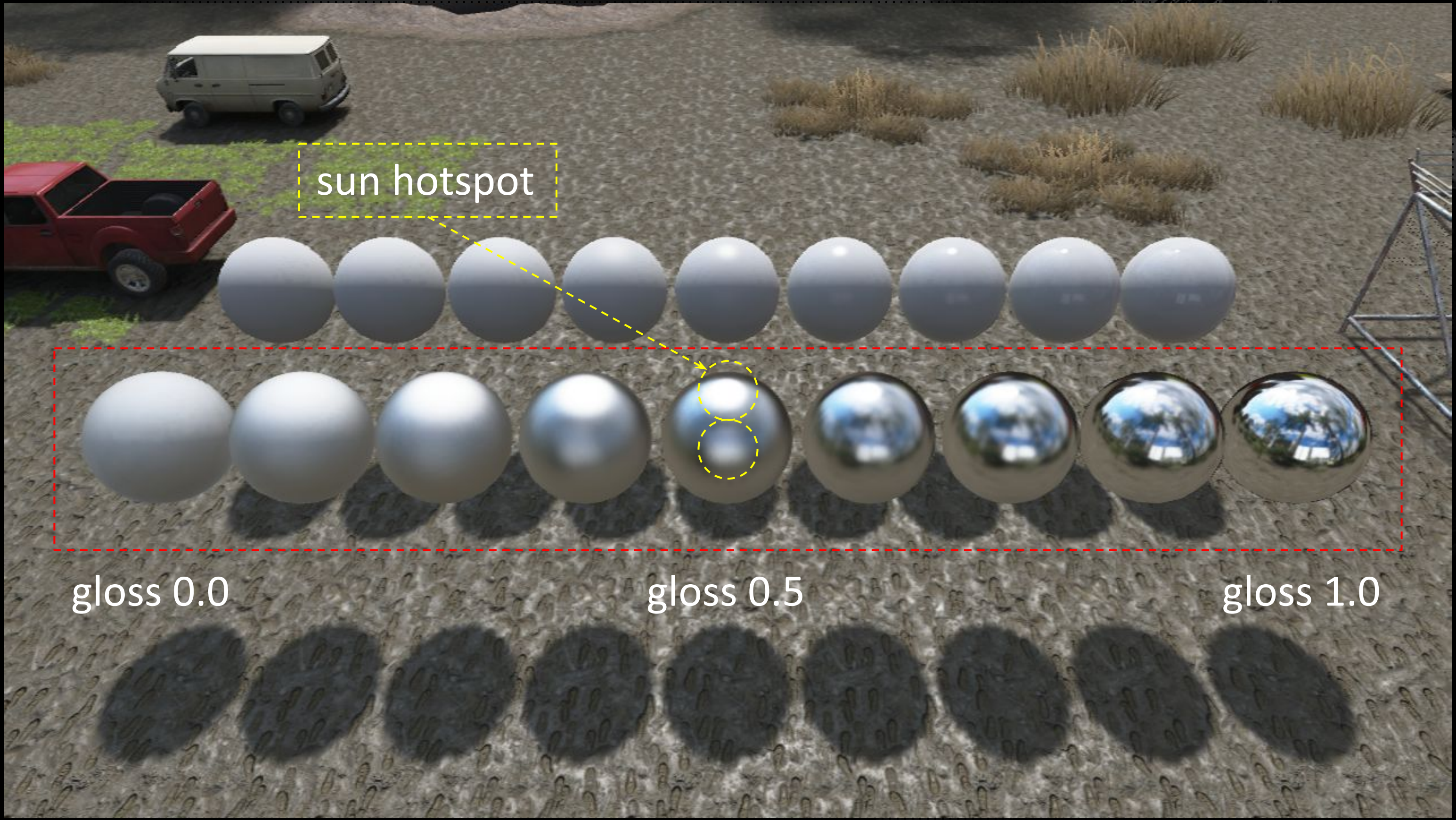
- Customized CubeMapGen with cosine power filter
  - Concurrent work with Sébastien Lagarde
- Each mip level filtered with matching gloss / specular power
- Top mip “resolution” tied to max specular power
  - Dropped environment map resolution from 256x256 to 128x128
- Blinn-Phong to Phong specular power conversion:

$$\alpha_{phong} = \alpha_{blinn\text{-}phong} / 4$$

Environment map pre-filtering: old method



Environment map pre-filtering: new method



Environment lighting: ground truth

- Environment lighting integral

$$\int Env(l) BRDF_{env}(l,v,h)\cos(\omega)d\omega$$

$$\downarrow$$

$$D_{env}(h)F(l,h)V(l,v,h) \leftarrow D_{env}(h) = \frac{\alpha+2}{8\pi}(n\cdot h)^{\alpha}$$

Environment lighting: split approximation

- Split the integral: easier to calculate the parts separately

$$\left(4\int Env(l)D_{env}(h)\cos(\omega)d\omega\right)\left(\int BRDF_{env}(l,v,h)\cos(\omega)d\omega\right)$$

Environment map filtering

↓

Approximate with mip map pre-filtering

Environment BRDF

(also referred to as "Ambient BRDF")

↓

Approximate with cheap analytical expressions

Environment BRDF: reflectance interpolation

- From the Fresnel formulation:

$$F(l,h) = rf_0 + (1 - rf_0)(1 - h \cdot l)^5$$

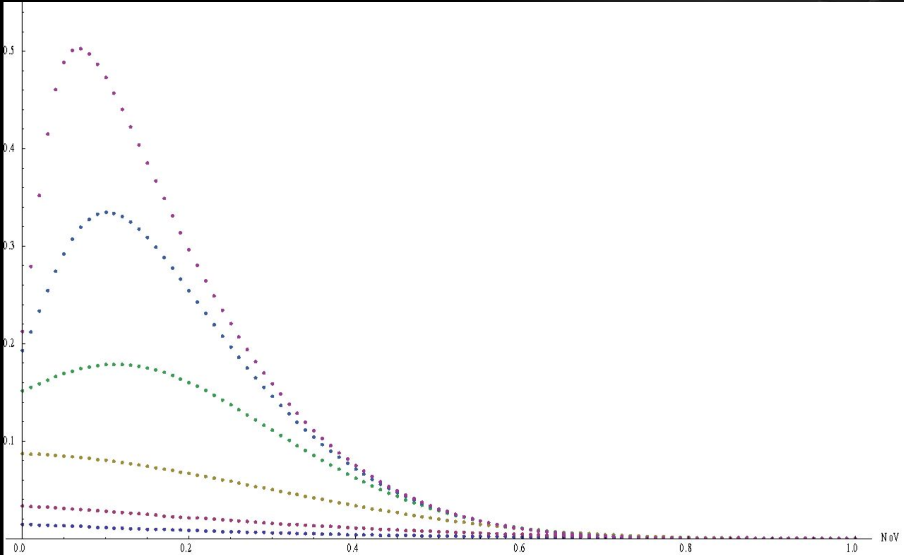
$$\int BRDF_{env} \cos(\omega) d\omega = rf_0 \int D_{env} V \cos(\omega) d\omega + (1 - rf_0) \int D_{env} V (1 - h \cdot l)^5 \cos(\omega) d\omega$$

$rf_0 = 1$

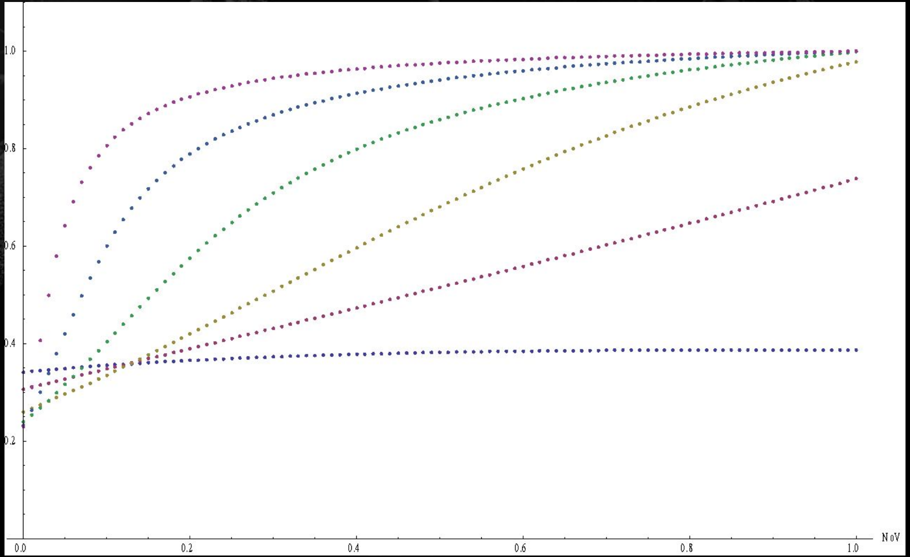
$rf_0 = 0$

Numerical integration in *Mathematica*

- Plotted two sets of ground-truth curves for $rf_0 = 0$ and $rf_0 = 1$
- Each set contained curves for gloss values 0.0, 0.2, 0.4, 0.6, 0.8 and 1.0

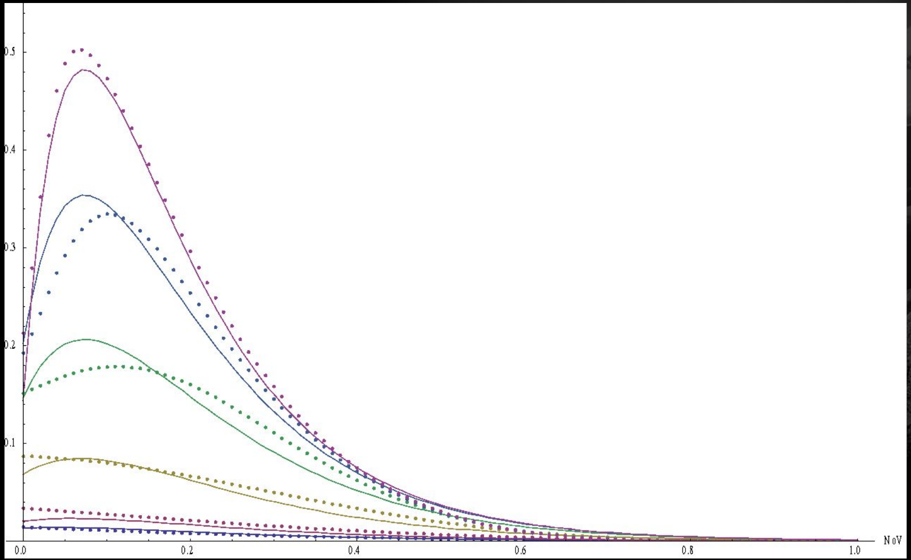


$rf_0 = 0$

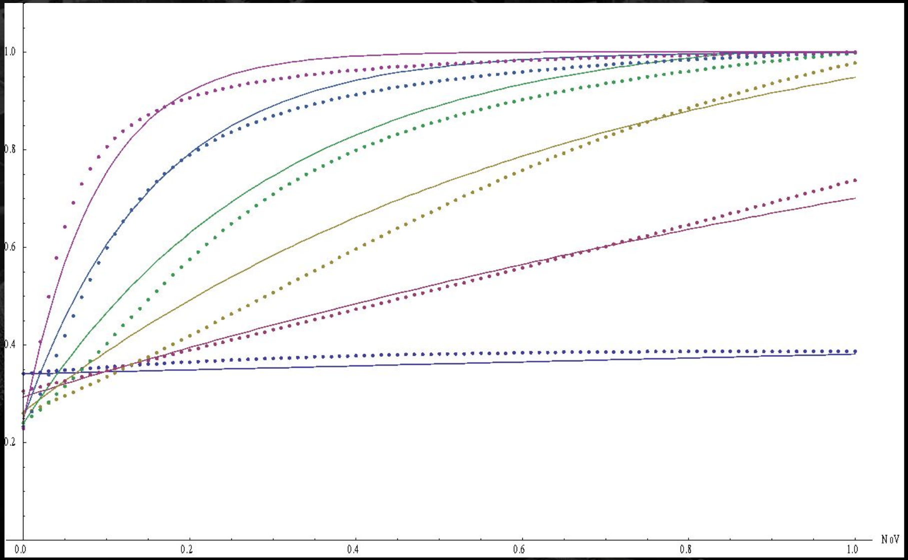


$rf_0 = 1$

Approximate curves: accurate



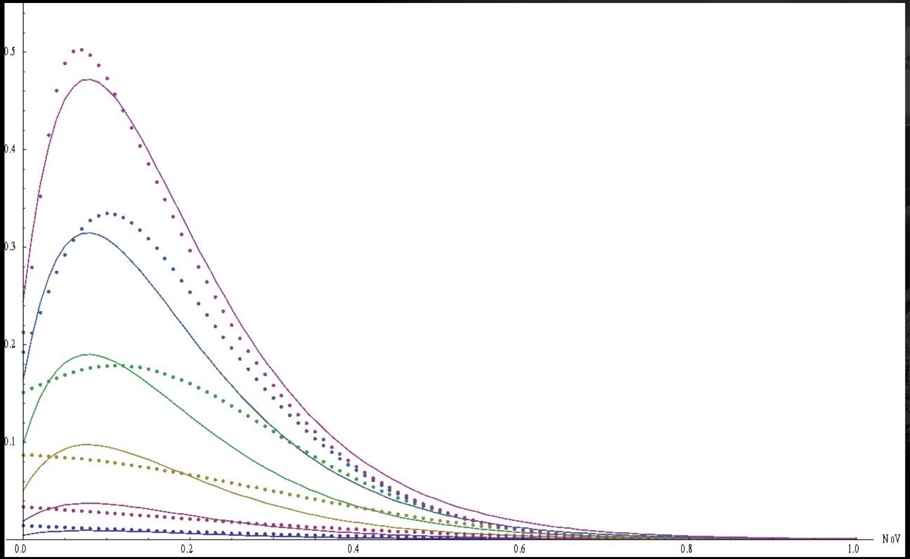
$rf_0 = 0$



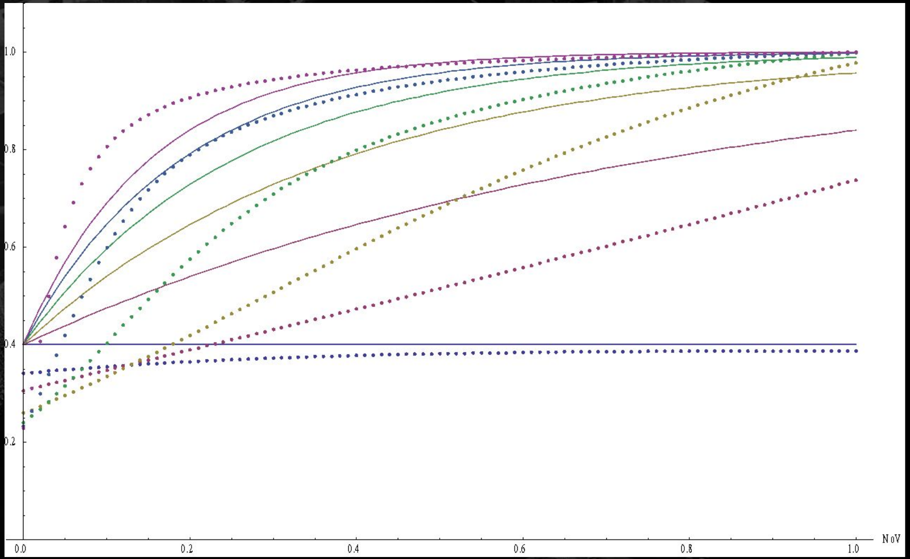
$rf_0 = 1$

* HLSL expressions in the course notes

Approximate curves: cheaper



$rf_0 = 0$



$rf_0 = 1$

* HLSL expressions in the course notes

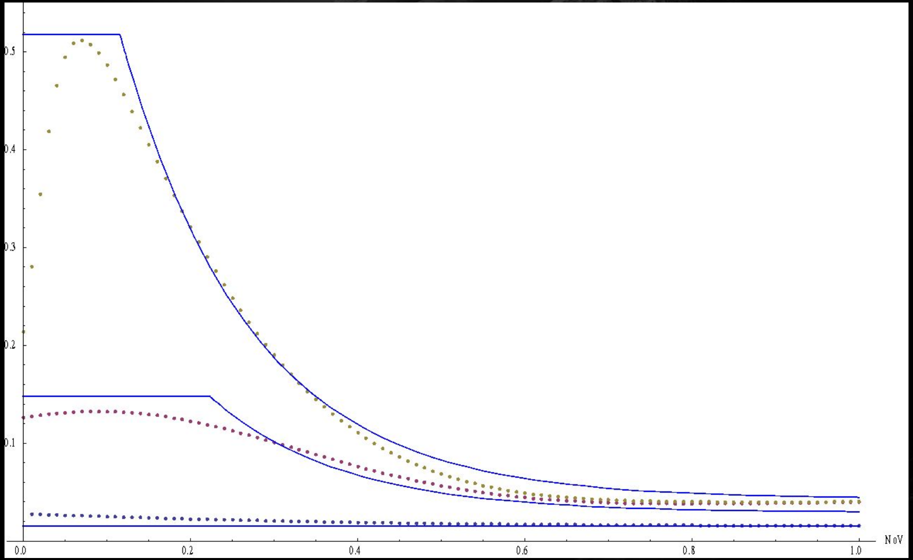
Focus on $rf_0 = 0.04$

- Needed faster approximations
- We had a special-case “simple” material (dielectric only) with a hardcoded specular color of 0.04
- Most of our environment specular problems revolved around dielectrics
- Metals looked good even with the cheapest approximations:

```
float a1vf(float g)
{
    return 0.25 * g + 0.75;
}
```

Approximate curves: $rf_0 = 0.04$

```
float a004(float g, float NoV)
{
    float t = min(0.475 * g, exp2(-9.28 * NoV));
    return (t + 0.0275) * g + 0.015;
}
```



g = 0.0, 0.5, 1.0

Final approximation

```
float a0r(float g, float NoV)
{
    return (a004(g, NoV) - a1vf(g) * 0.04) / 0.96;
}

float3 EnvironmentBRDF(float g, float NoV, float3 rf0)
{
    float4 t = float4(1/0.96, 0.475, (0.0275 - 0.25*0.04)/0.96, 0.25);
    t *= float4(g, g, g, g);
    t += float4(0, 0, (0.015 - 0.75*0.04)/0.96, 0.75);
    float a0 = t.x * min(t.y, exp2(-9.28 * NoV)) + t.z;
    float a1 = t.w;
    return saturate(a0 + rf0 * (a1 - a0));
}
```

Environment BRDF: old method

Environment BRDF: new method

Acknowledgments

- Naty Hoffman
- Marc Olano
- Jorge Jimenez
- Sébastien Lagarde
- Stephen Hill & Stephen McAuley
- The team at Treyarch

We are hiring

- You can find a list of our open positions at www.activisionblizzard.com/careers. Here is just a sample of what Treyarch currently has available:

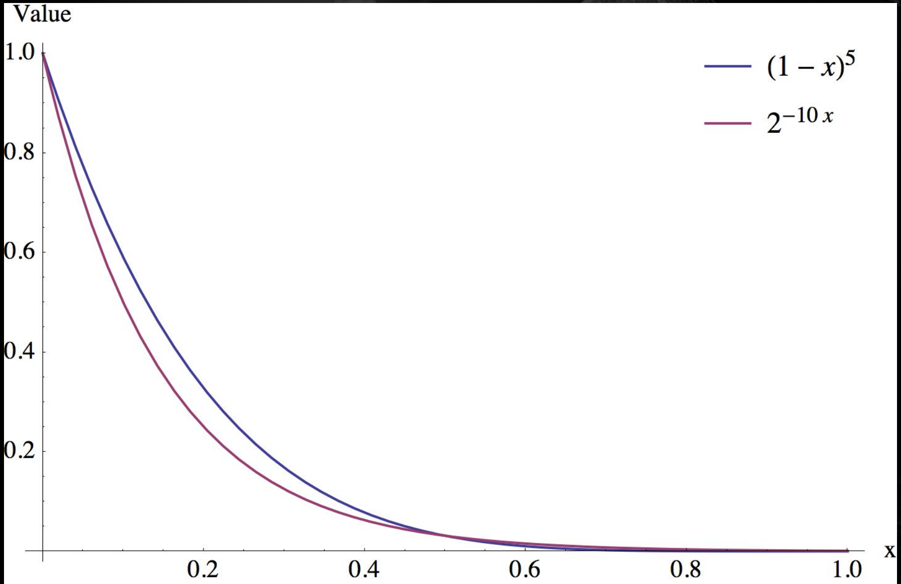
- Senior Graphics Engineer
- Senior Concept Artist-Vehicles/Weapons
- Senior Artist-Vehicles/Weapons
- Technical Animator

Bonus slides

Black Ops II: new Fresnel approximation

- Used *Mathematica* to fit candidate curves

$$F_{opt}(l,h) = rf_0 + (1 - rf_0)2^{-10(h \cdot l)}$$

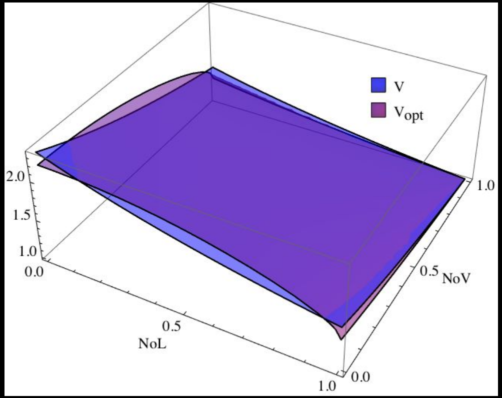


Black Ops II: new visibility function approximation

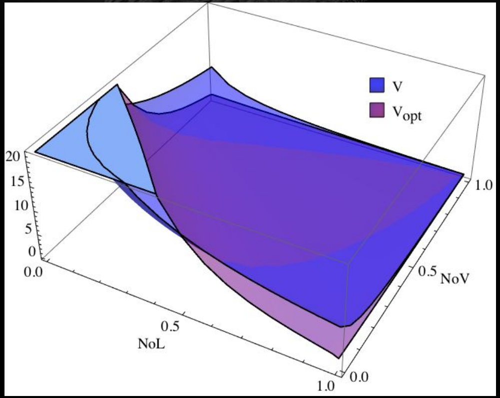
- Visually matched in game
  (not an exact fit, but much faster)

$$k = \min(1.0, g + 0.545)$$

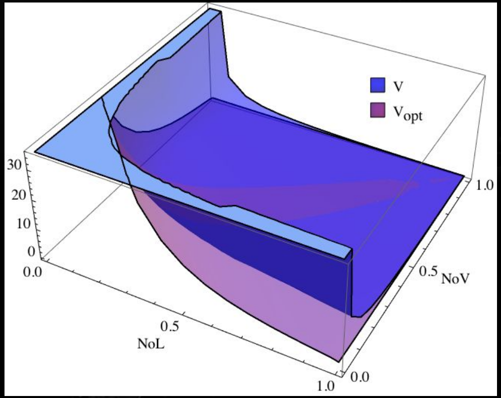
$$V_{opt}(v,h) = \frac{1}{k(v \cdot h)^2 + (1-k)}$$



g = 0.0



g = 0.5



g = 1.0