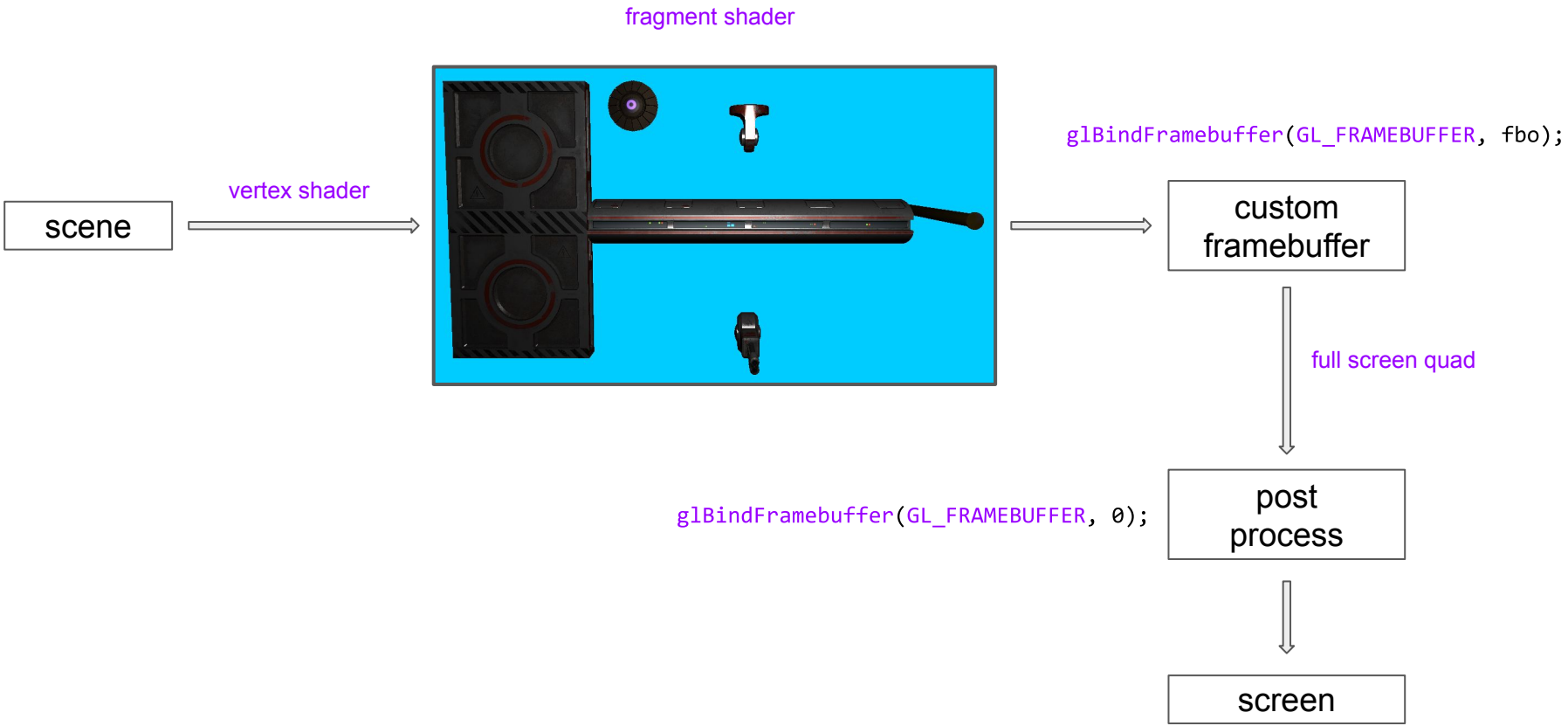
Deferred rendering
&
Image-based lighting

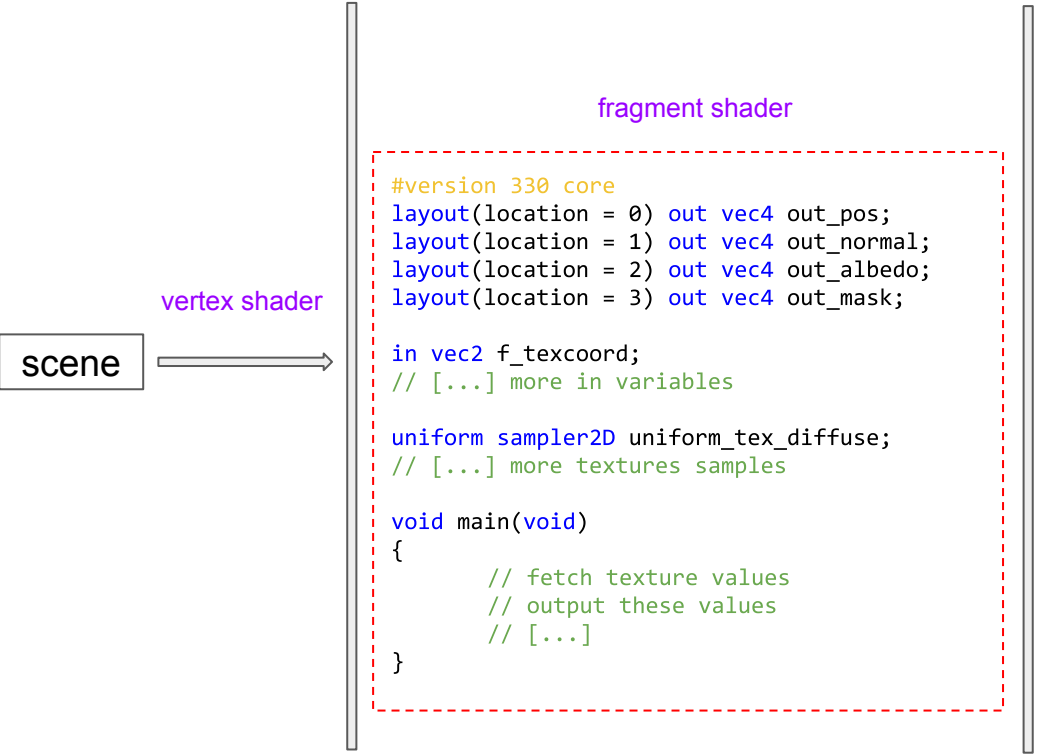
Evangelou Iordanis

Forward shading



Deferred shading

```
glBindFramebuffer(GL_FRAMEBUFFER, fbo);
```



Deferred shading

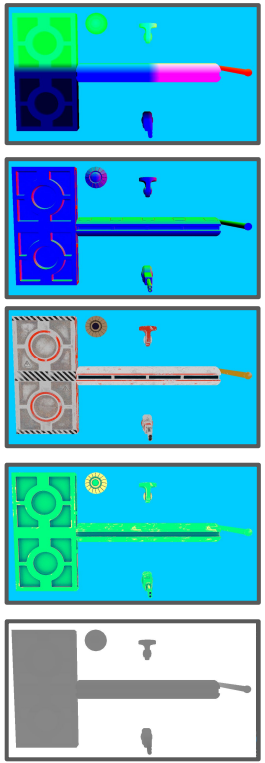
```
glBindFramebuffer(GL_FRAMEBUFFER, fbo);
```

scene

vertex shader

fragment shader

```
#version 330 core
layout(location = 0) out vec4 out_pos;
layout(location = 1) out vec4 out_normal;
layout(location = 2) out vec4 out_albedo;
layout(location = 3) out vec4 out_mask;

in vec2 f_texcoord;
// [...] more in variables

uniform sampler2D uniform_tex_diffuse;
// [...] more textures samples

void main(void)
{
        // fetch texture values
        // output these values
        // [...]
}
```



Deferred shading

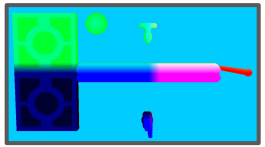
```
glBindFramebuffer(GL_FRAMEBUFFER, fbo);
```

scene

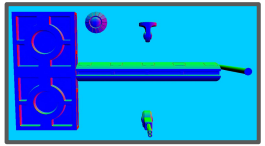
vertex shader

fragment shader

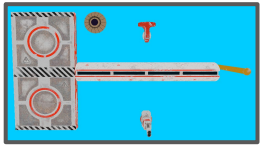
```
#version 330 core
layout(location = 0) out vec4 out_pos;
layout(location = 1) out vec4 out_normal;
layout(location = 2) out vec4 out_albedo;
layout(location = 3) out vec4 out_mask;

in vec2 f_texcoord;
// [...] more in variables

uniform sampler2D uniform_tex_diffuse;
// [...] more textures samples

void main(void)
{
        // fetch texture values
        // output these values
        // [...]
}
```



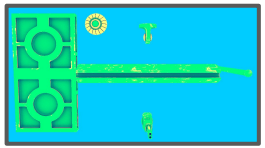
```
glFramebufferTexture2D(GL_FRAMEBUFFER,
        GL_COLOR_ATTACHMENT0,
        GL_TEXTURE_2D, fbo_pos_texture, 0);
```



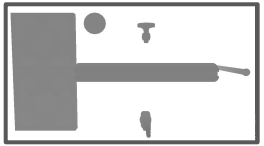
```
glFramebufferTexture2D(GL_FRAMEBUFFER,
        GL_COLOR_ATTACHMENT1,
        GL_TEXTURE_2D, fbo_normal_texture, 0);
```



```
glFramebufferTexture2D(GL_FRAMEBUFFER,
        GL_COLOR_ATTACHMENT2,
        GL_TEXTURE_2D, fbo_albedo_texture, 0);
```



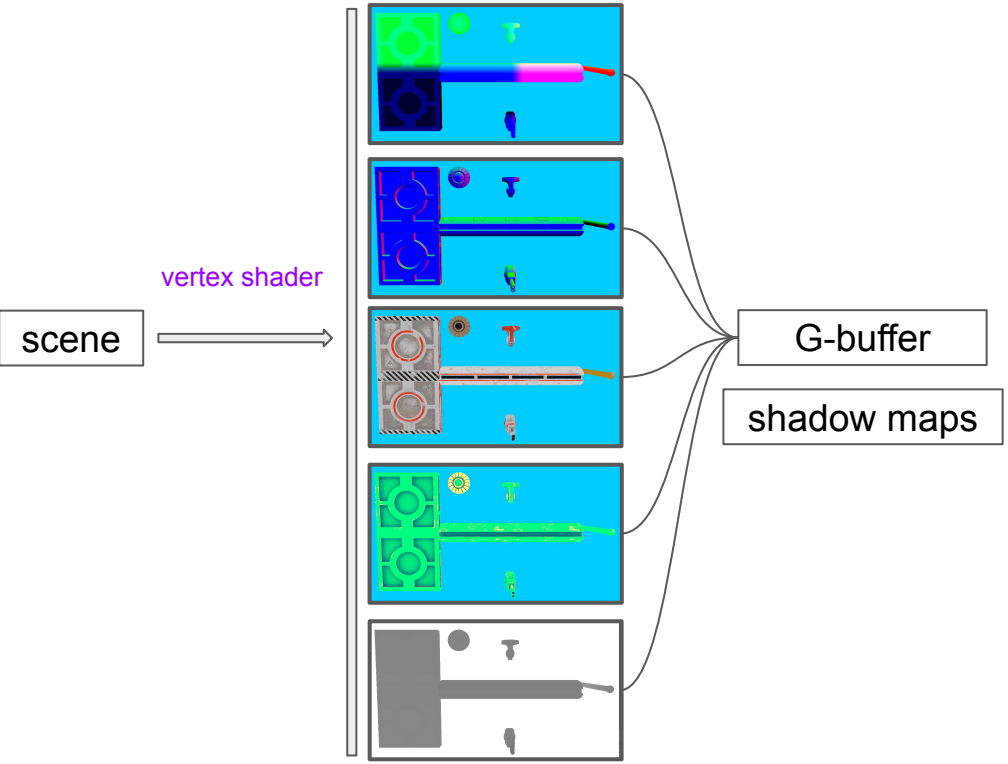
```
glFramebufferTexture2D(GL_FRAMEBUFFER,
        GL_COLOR_ATTACHMENT3,
        GL_TEXTURE_2D, fbo_mask_texture, 0);
```



```
glFramebufferTexture2D(GL_FRAMEBUFFER,
        GL_DEPTH_COMPONENT,
        GL_TEXTURE_2D, fbo_depth_texture, 0);
```

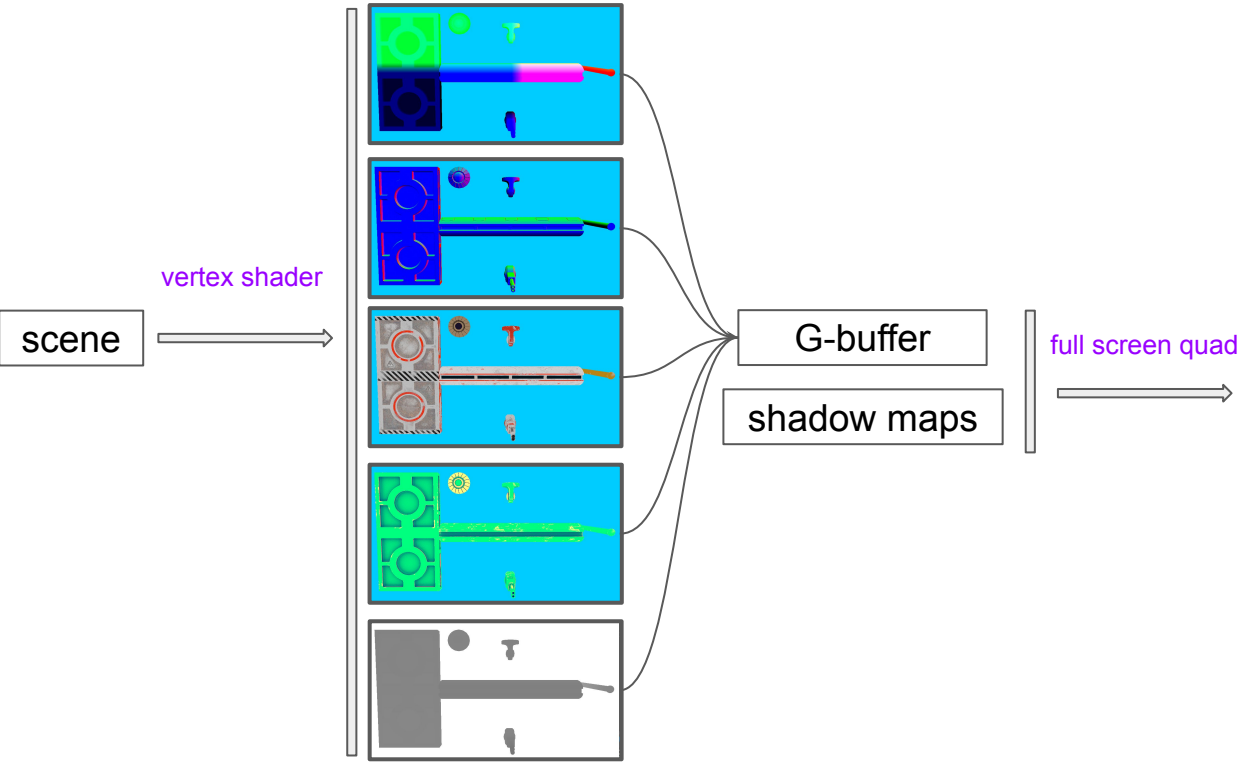
Deferred shading

```
glBindFramebuffer(GL_FRAMEBUFFER, fbo);
```



Deferred shading

```
glBindFramebuffer(GL_FRAMEBUFFER, fbo);
```

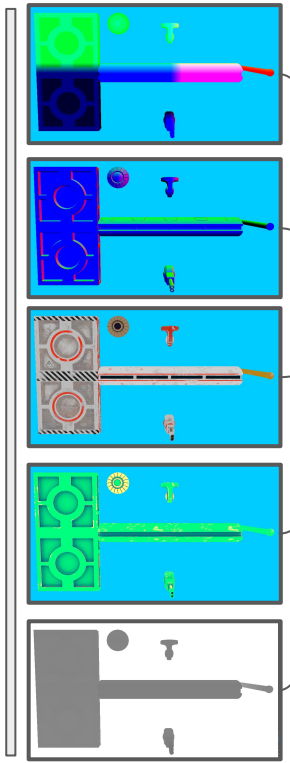


fragment shader

```
#version 330 core
layout(location = 0) out vec4 out_color;

in vec2 f_texcoord;

// G-buffer samplers
uniform sampler2D uniform_tex_pos;


// [...] more textures samplers / uniforms

void main(void)
{
        // evaluate lighting
        // output shading to out_color
}
```

Deferred shading

```
glBindFramebuffer(GL_FRAMEBUFFER, fbo);
```



vertex shader

scene

G-buffer

shadow maps

full screen quad

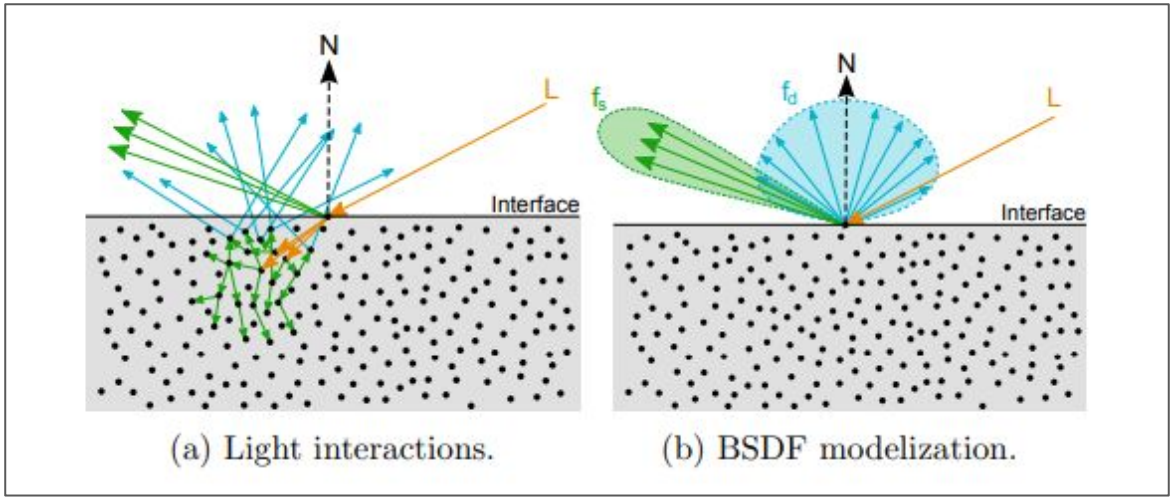
full screen quad

```
glBindFramebuffer(GL_FRAMEBUFFER, 0);
```

post process

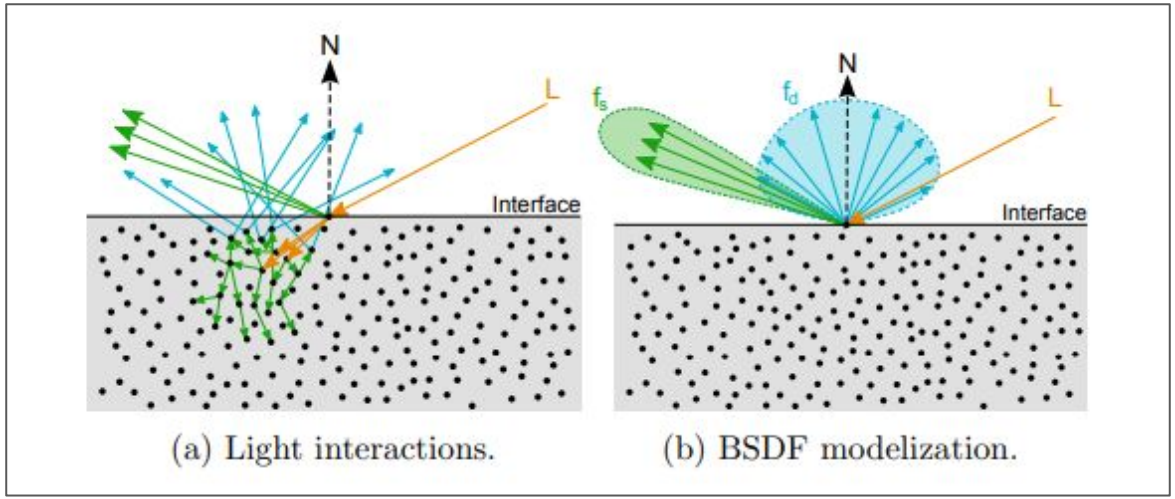
screen

Image-Based Lighting



(a) Light interactions. (b) BSDF modelization.

Image-Based Lighting

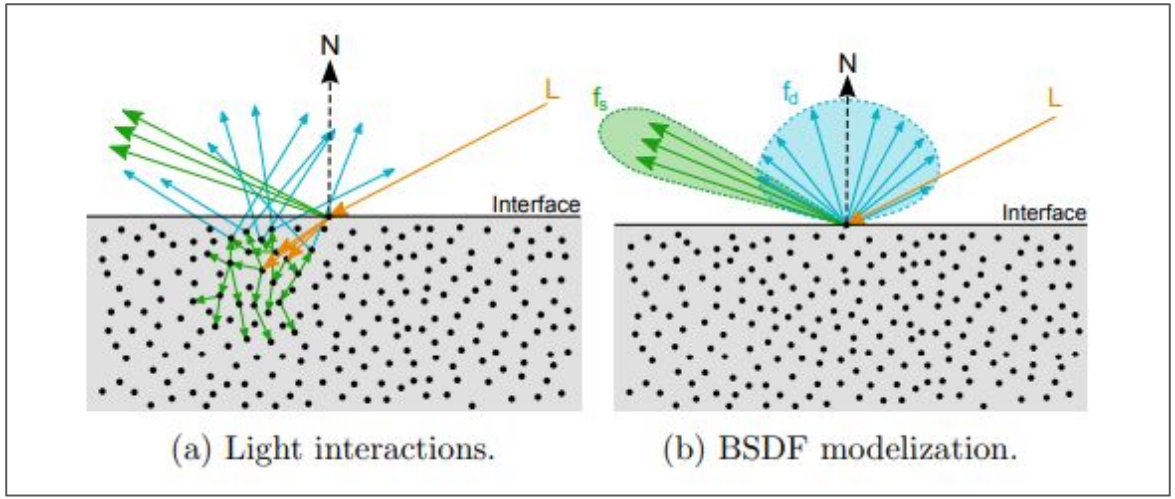


(a) Light interactions. (b) BSDF modelization.

$$L(p, \omega_o) = L_e + \int_H f(p, \omega_o, \omega_i)\, L(p, \omega_i)\, \cos\theta\, d_{\omega_i}$$

- integral over all possible directions

Image-Based Lighting

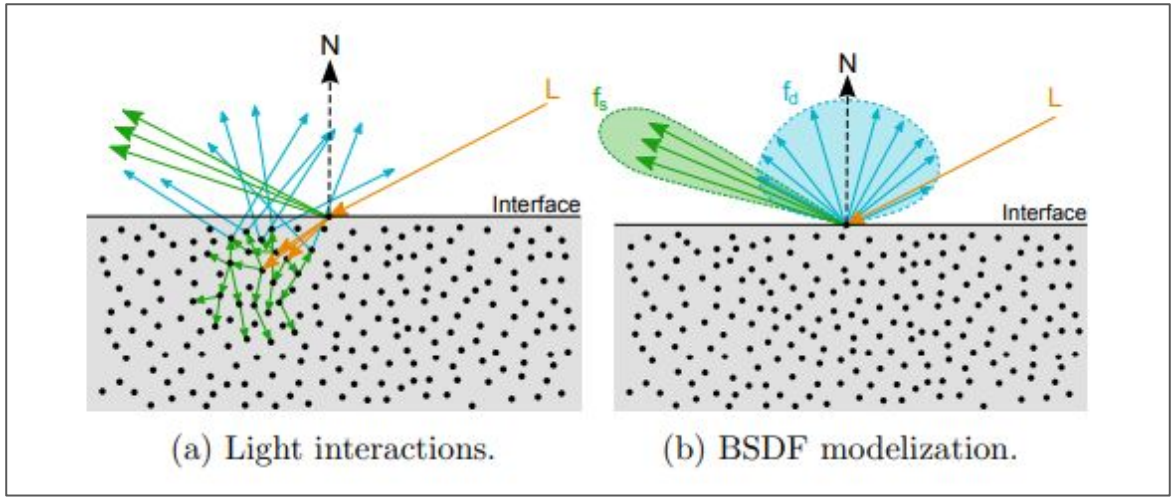


(a) Light interactions. (b) BSDF modelization.

$$L(p, \omega_o) = L_e + \int_H f(p, \omega_o, \omega_i) L(p, \omega_i) \cos\theta \, d_{\omega_i}$$

- integral over all possible directions
- BRDF based on cook-torrance model

Image-Based Lighting

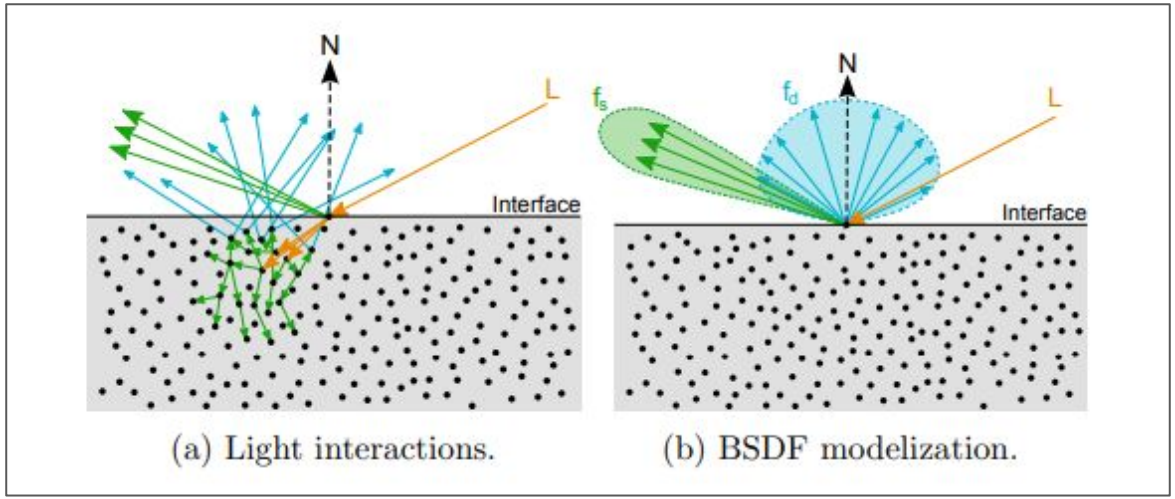


(a) Light interactions. (b) BSDF modelization.

$$L(p, \omega_o) = L_e + \int_H f(p, \omega_o, \omega_i) L(p, \omega_i) \cos\theta \, d_{\omega_i}$$

- integral over all possible directions
- BRDF based on cook-torrance model
- Incident radiance

Image-Based Lighting

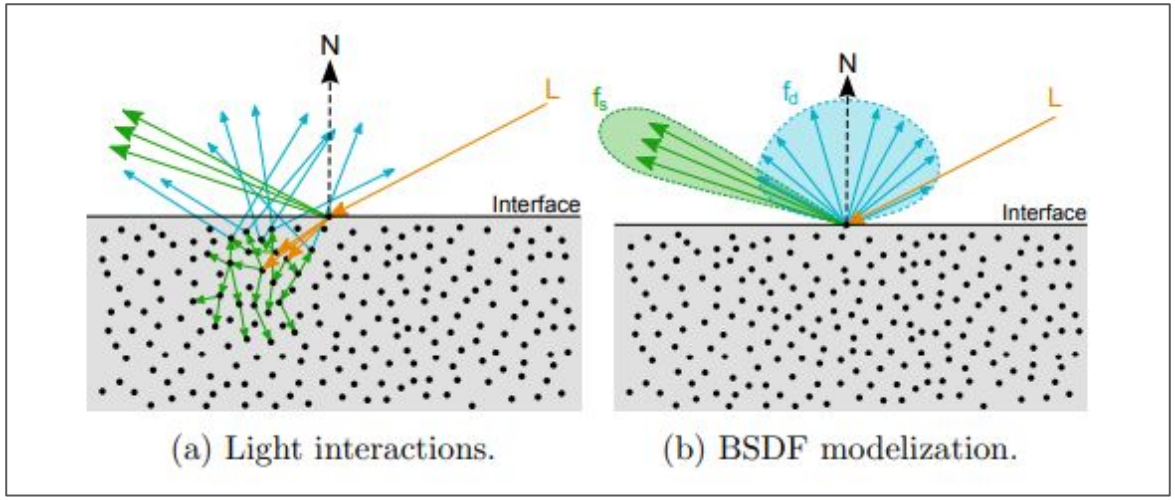


(a) Light interactions. (b) BSDF modelization.

$$L(p, \omega_o) = L_e + \int_H f(p, \omega_o, \omega_i)\, L(p, \omega_i) \cos\theta\, d_{\omega_i}$$

- integral over all possible directions
- BRDF based on cook-torrance model
- incident radiance
- “energy” scaler

Image-Based Lighting



(a) Light interactions. (b) BSDF modelization.

$$L(p, \omega_o) = L_e + \int_H f(p, \omega_o, \omega_i)\, L(p, \omega_i)\, \cos\theta\, d_{\omega_i}$$

- integral over all possible directions
- BRDF based on cook-torrance model
- incident radiance
- “energy” scaler
- surface emission

Image-Based Lighting

Reflectance equation :

$$L(p, \omega_o) = L_e + \int_H f(p, \omega_o, \omega_i)\, L(p, \omega_i)\, \cos\theta\, d_{\omega_i}$$

- for direct illumination we need to evaluate the latter for every light, hence :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} \int_H f(p, \omega_o, \omega_i)\, L_i(p, \omega_i)\, \cos\theta\, d_{\omega_i}$$

Image-Based Lighting

Reflectance equation :

$$L(p, \omega_o) = L_e + \int_H f(p, \omega_o, \omega_i)\, L(p, \omega_i)\, \cos\theta\, d_{\omega_i}$$

- for direct illumination we need to evaluate the latter for every light, hence :

  $$L(p, \omega_o) = L_e + \sum_{i=1}^{n} \int_H f(p, \omega_o, \omega_i)\, L_i(p, \omega_i)\, \cos\theta\, d_{\omega_i}$$

- and since we only have point/spot lights :

  $$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i)\, L_i(p, \omega_i)\, \cos\theta$$

Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos\theta$$

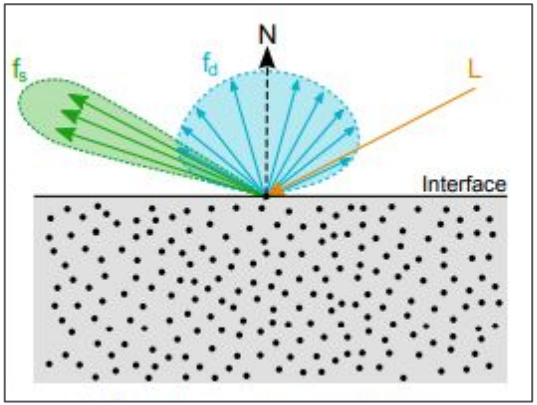
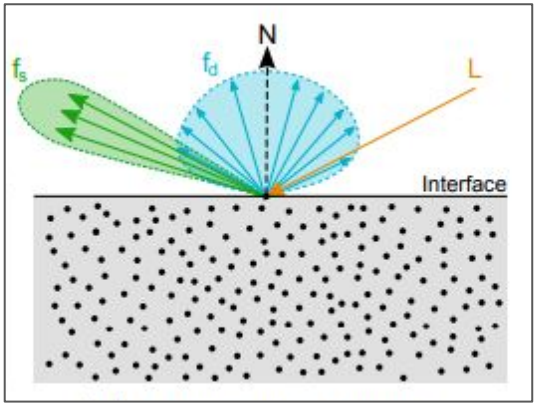


Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos\theta$$



- the Cook-Torrance BRDF model is :

$$f(p, \omega_o, \omega_i) = (1 - F) f_d + F f_s$$
$$= (1 - F) \frac{c}{\pi} + F \frac{DG}{4(n \cdot l)(n \cdot v)}$$

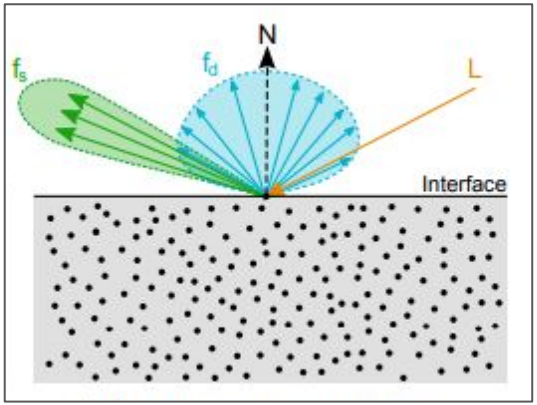
Lambertian isotropic

Specular/glossy reflection

Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos\theta$$



- the Cook-Torrance BRDF model is :

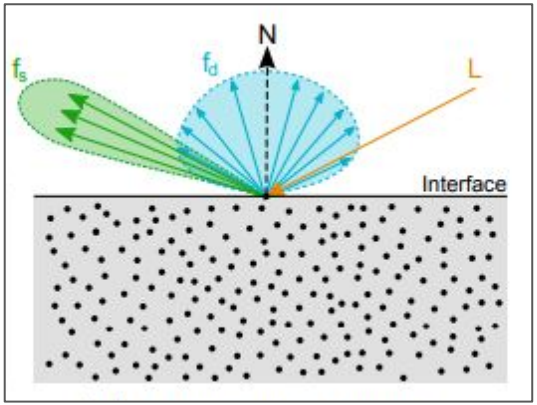
$$f(p, \omega_o, \omega_i) = (1 - F)\frac{c}{\pi} + F\frac{DG}{4(n \cdot l)(n \cdot v)}$$

probability of reflection

Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos\theta$$



- the Cook-Torrance BRDF model is :

$$f(p, \omega_o, \omega_i) = (1 - F)\frac{c}{\pi} + F\frac{DG}{4(n \cdot l)(n \cdot v)}$$

probability of reflection

- Fresnel term based on Schlick's approximation :

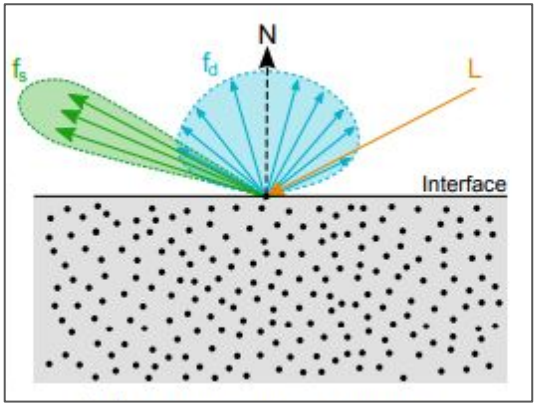
$$F(h, v, F_0) = F_0 + (1 - F_0)(1 - (h \cdot v))^5$$

reflectance at normal incidence

Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos\theta$$



- the Cook-Torrance BRDF model is :

$$f(p, \omega_o, \omega_i) = (1 - F)\frac{c}{\pi} + F\frac{DG}{4(n \cdot l)(n \cdot v)}$$

probability of reflection

- Fresnel term based on Schlick's approximation :

$$F(h, v, F_0) = F_0 + (1 - F_0)(1 - (h \cdot v))^5$$

reflectance at normal incidence

- and to account for the metallic term :

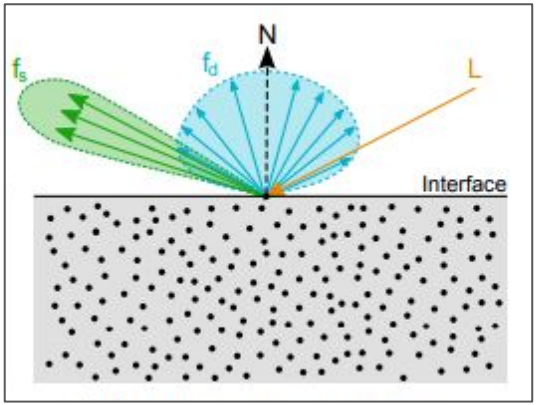
$$F_0 = (1 - \text{metallic})\, r + (\text{metallic}) albedo$$

$$f_d = (1 - \text{metallic})\frac{c}{\pi}$$

Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos\theta$$



- the Cook-Torrance BRDF model is :

$$f(p, \omega_o, \omega_i) = (1 - F)\frac{c}{\pi} + F\frac{DG}{4(n \cdot l)(n \cdot v)}$$

- microfacet distribution function :

$$D(h) = \frac{\alpha^2}{\pi\Big((n \cdot h)^2(\alpha^2 - 1) + 1\Big)^2}$$

- microfacet geometric function :

$$G(h) = \min\left\{1, \frac{2(n \cdot h)(n \cdot v)}{v \cdot h}, \frac{2(n \cdot h)(n \cdot l)}{v \cdot h}\right\}$$

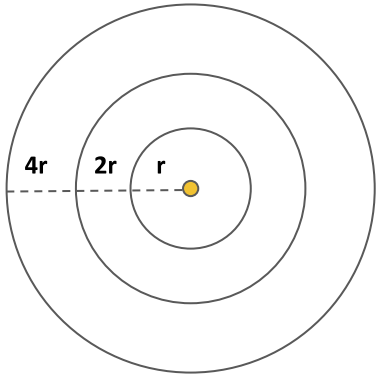
Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos\theta$$

- incoming radiant flux
  - for point lights :

$$I(p) = \frac{flux}{4\pi r^2}, \quad r = \|p - l\|$$



  - for spot lights :

$$I(p) = \frac{flux}{2\pi r^2 \left(1 - \frac{1}{2}(\cos\theta_1 + \cos\theta_2)\right)}$$

$$r = \|p - l\|$$

$$\cos\theta_1 = \text{umbra}$$

$$\cos\theta_2 = \text{penumbra}$$

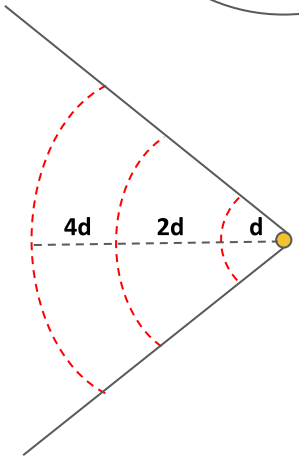


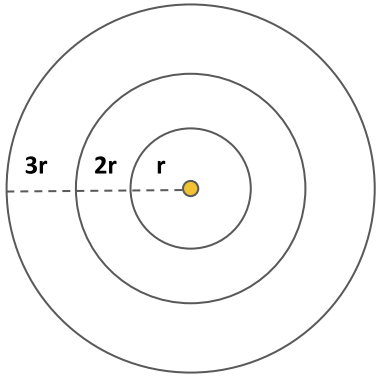
Image-Based Lighting

Reflectance equation for direct lighting :

$$L(p, \omega_o) = L_e + \sum_{i=1}^{n} f(p, \omega_o, \omega_i)\, L_i(p, \omega_i) \cos\theta$$

- incoming radiant flux
  - for point lights :

$$I(p) = \frac{flux}{4\pi r^2}, \quad r = \|p - l\|$$



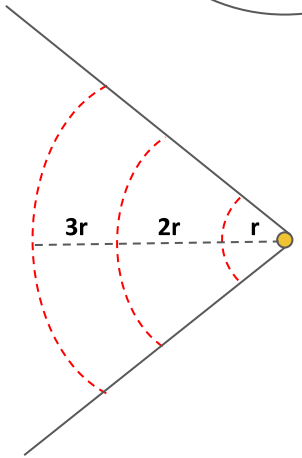
  - for spot lights :

$$I(p) = \frac{flux}{2\pi r^2\left(1 - \frac{1}{2}(\cos\theta_1 + \cos\theta_2)\right)}$$

$$r = \|p - l\|$$

$$\cos\theta_1 = \text{umbra}$$

$$\cos\theta_2 = \text{penumbra}$$



- “energy” scaler :

$$\cos\theta = n \cdot l$$