Computer Graphics

- A Primer on Rendering -

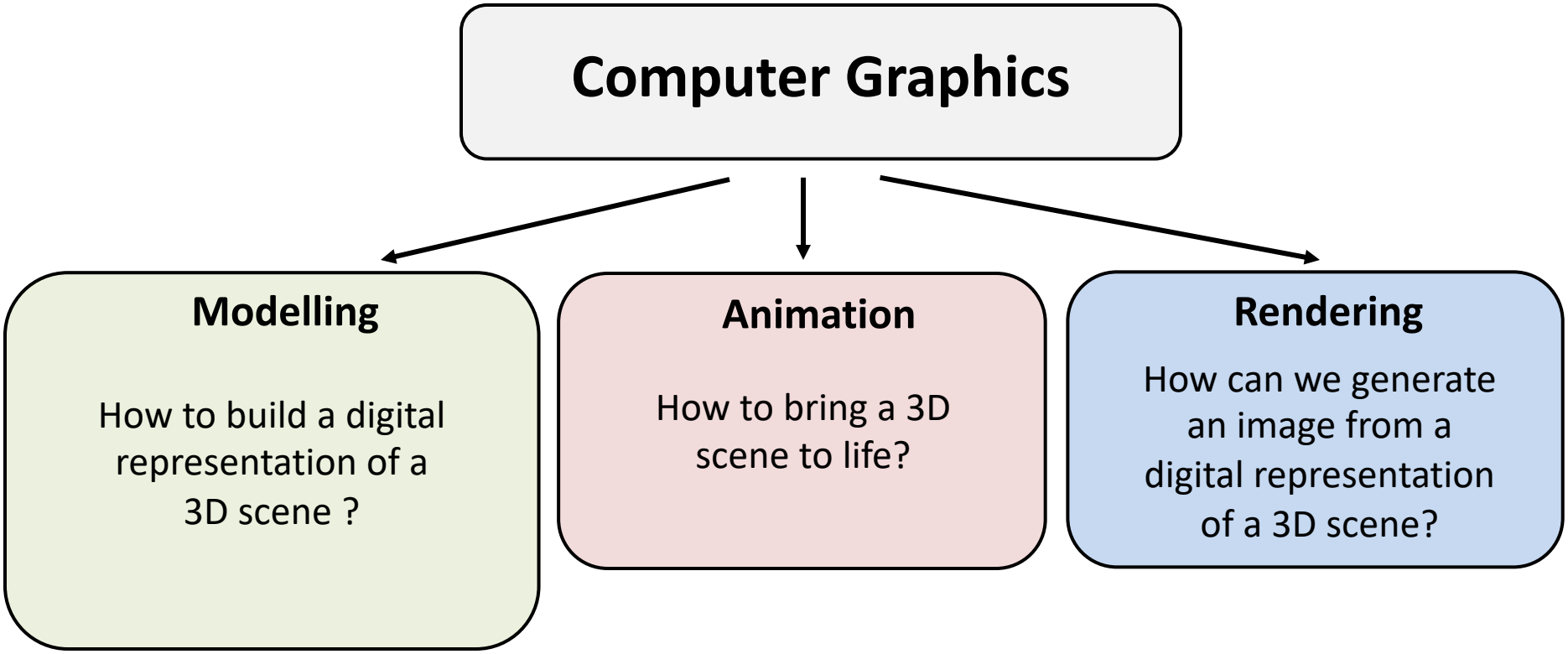
Philippe Weier

Alexander Rath

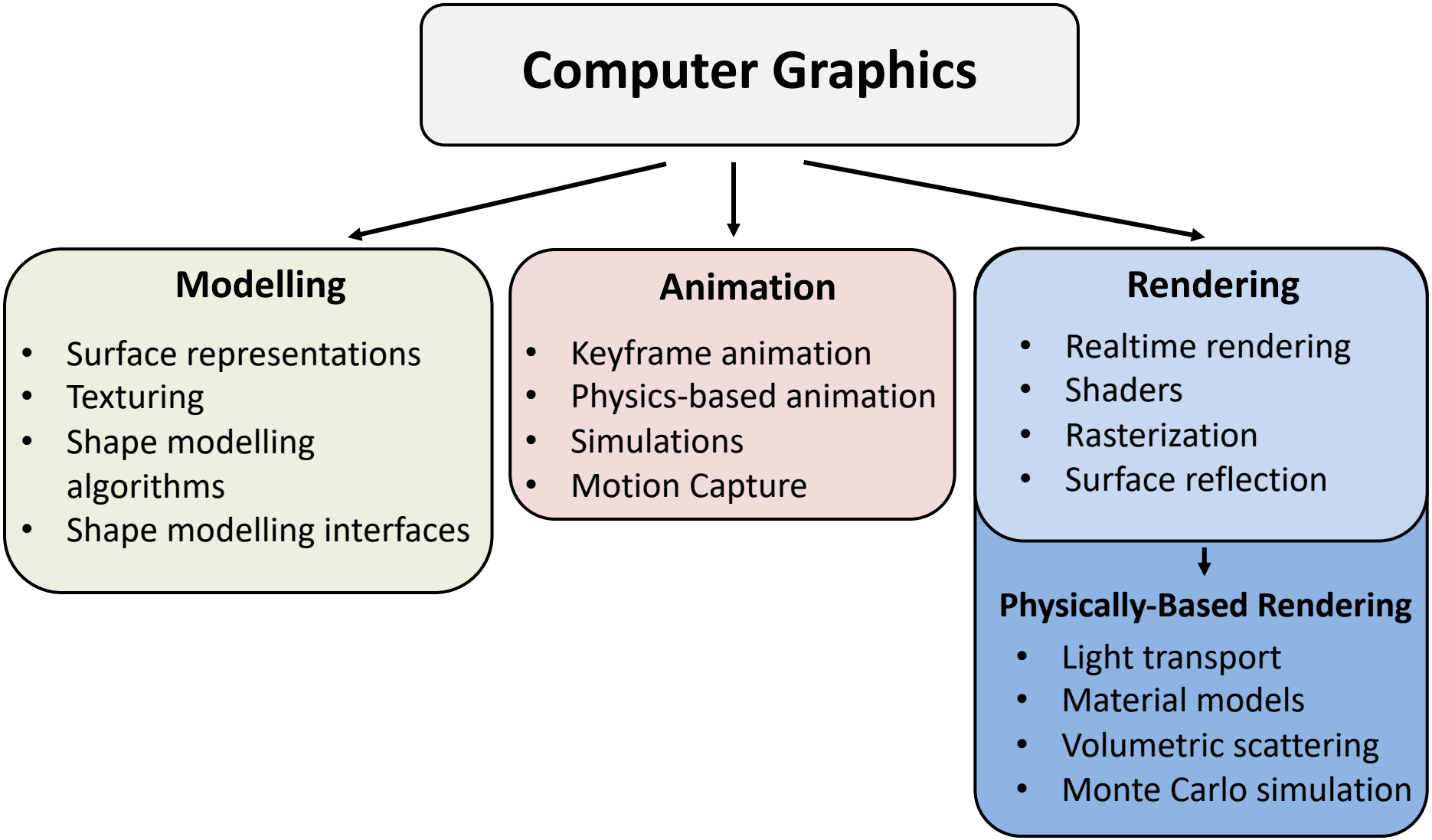
Philipp Slusallek

(many slides inspired by Prof. Wenzel Jakob)

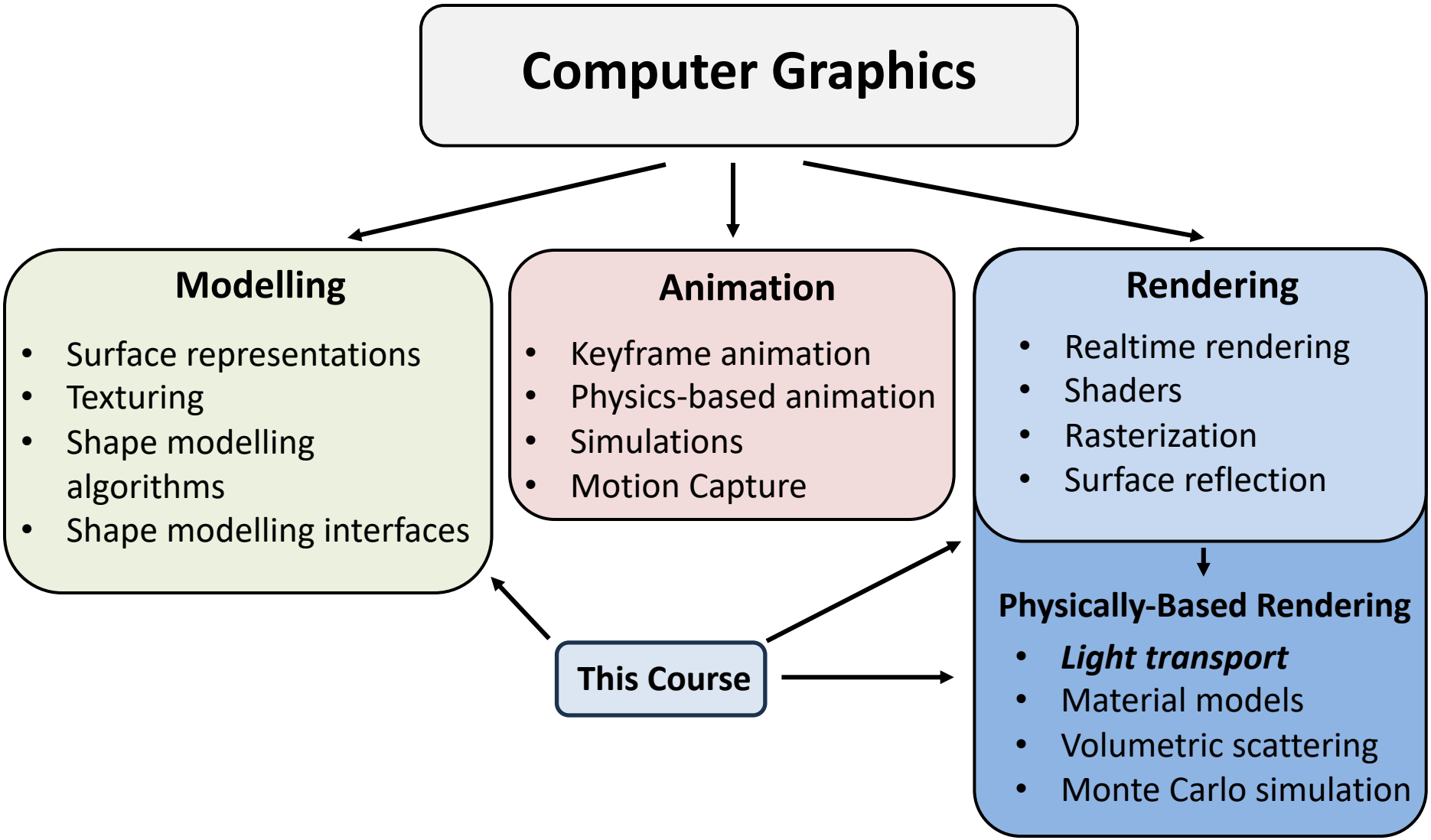
What is Rendering?



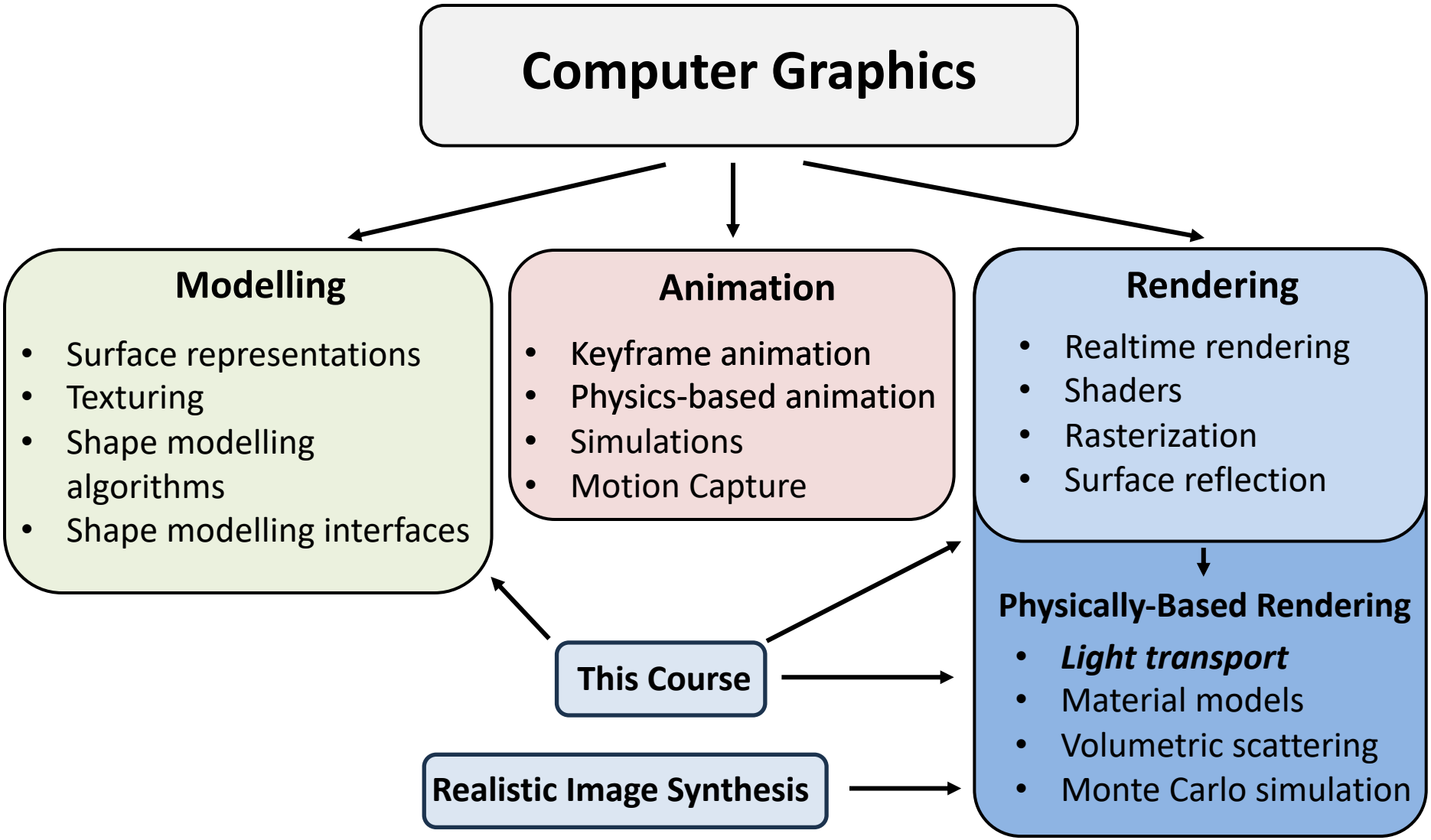
What is Rendering?



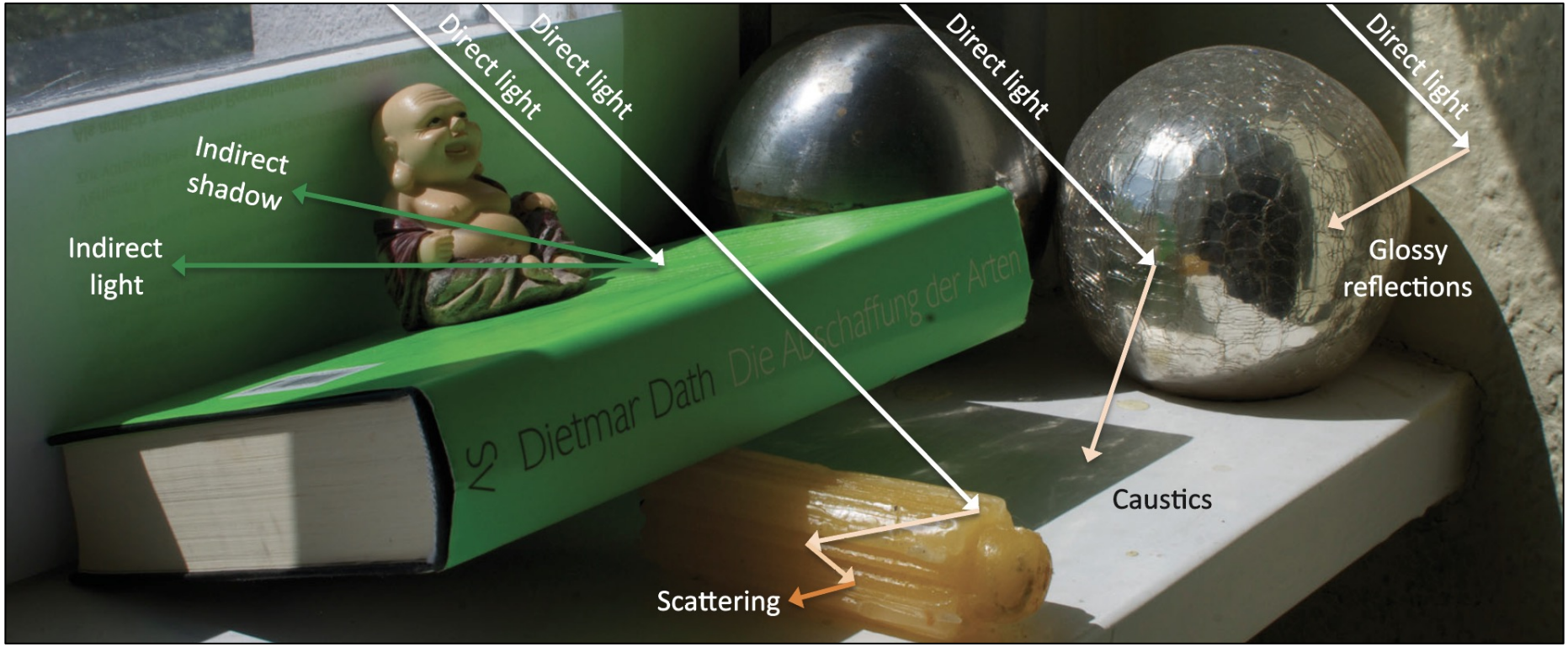
What is Rendering?



What is Rendering?



Light transport in real life



After [Ritschel et. al 2011]

Light transport in real life

Caustics

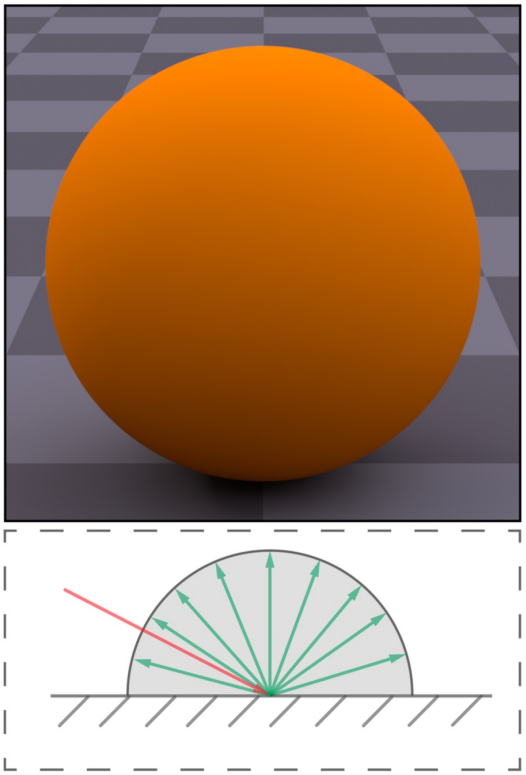
Light transport in real life

Subsurface Scattering

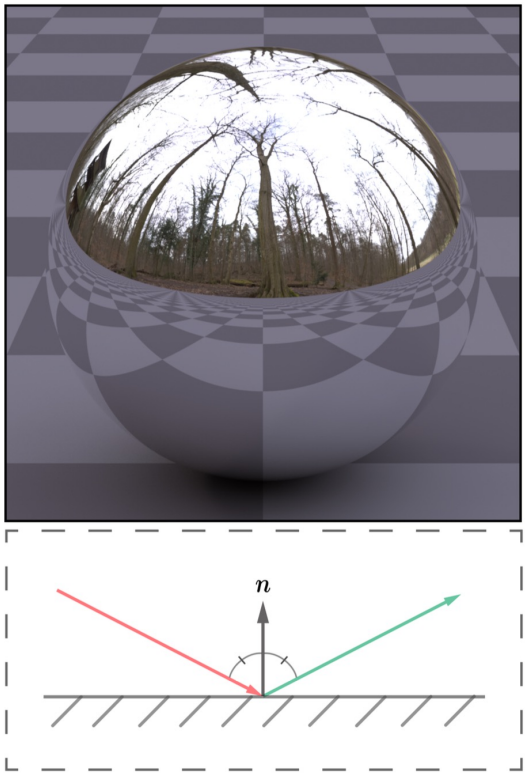
Light transport in real life

Participating Mediums (Volumetric Rendering)

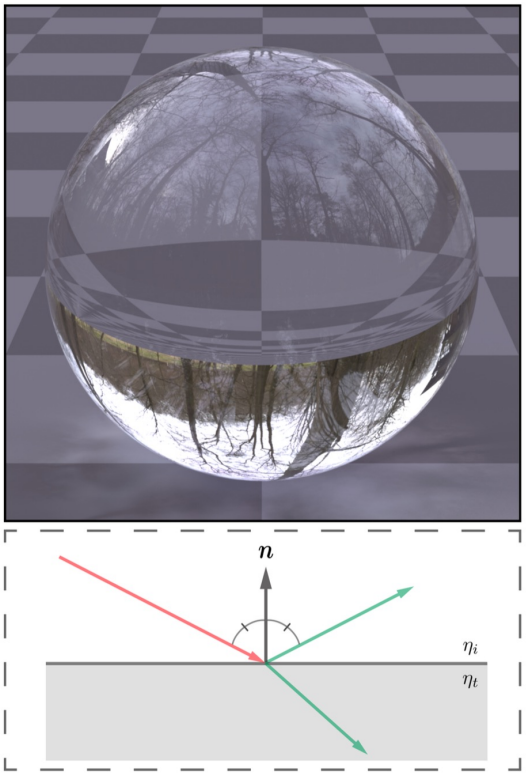
Material models



A diffuse surface

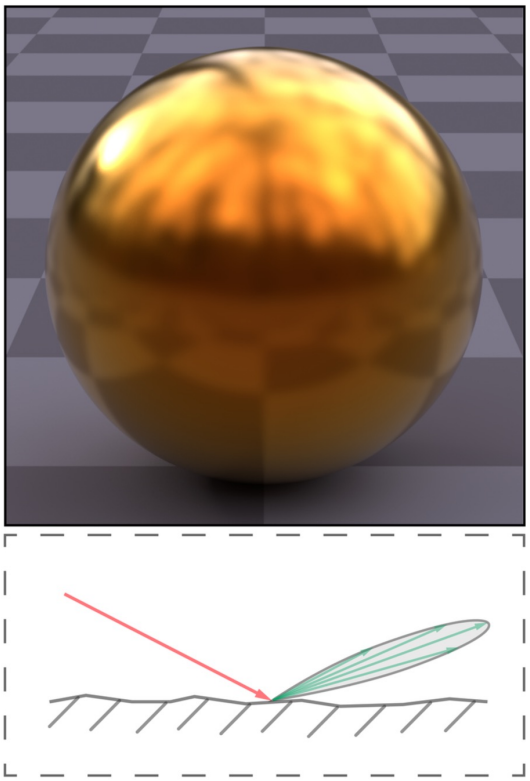


A conductor

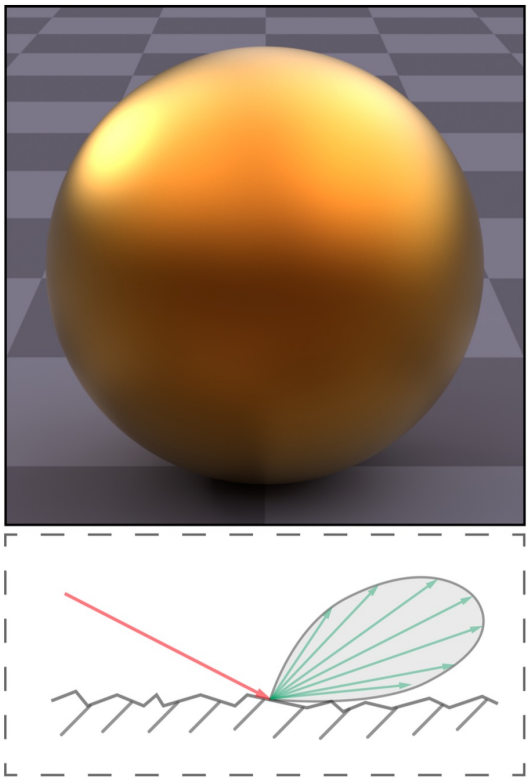


A dielectric

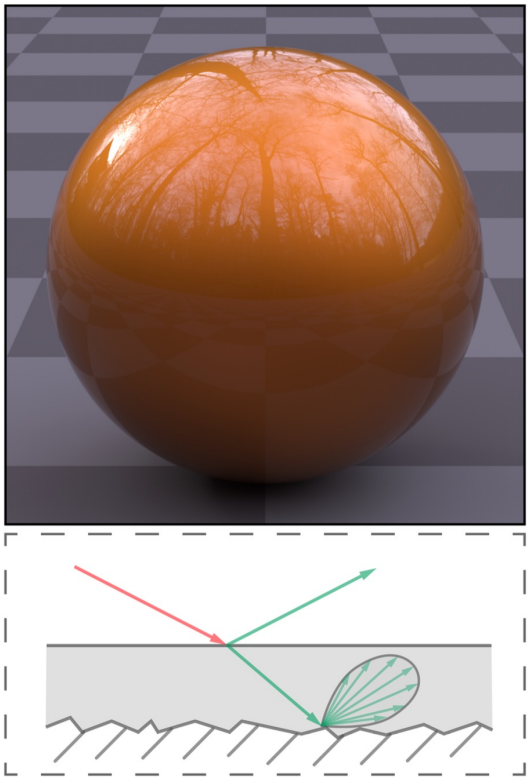
Material models



An almost specular surface

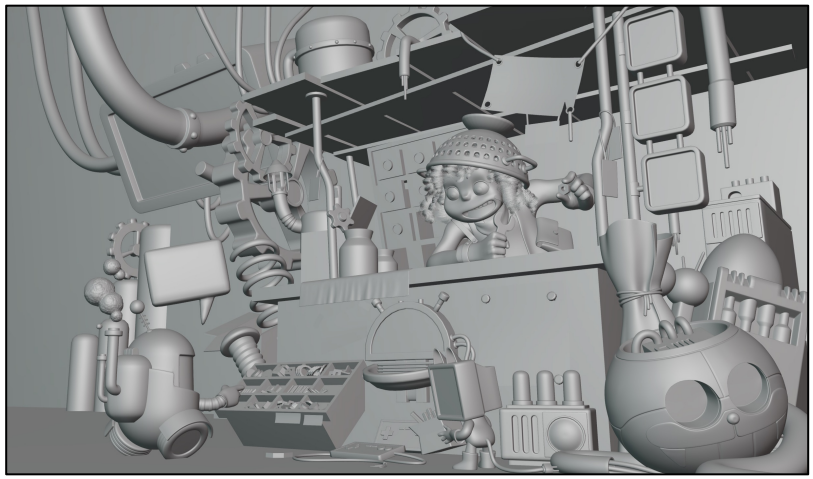


A rough surface



A multi-layered material

Rendering a scene



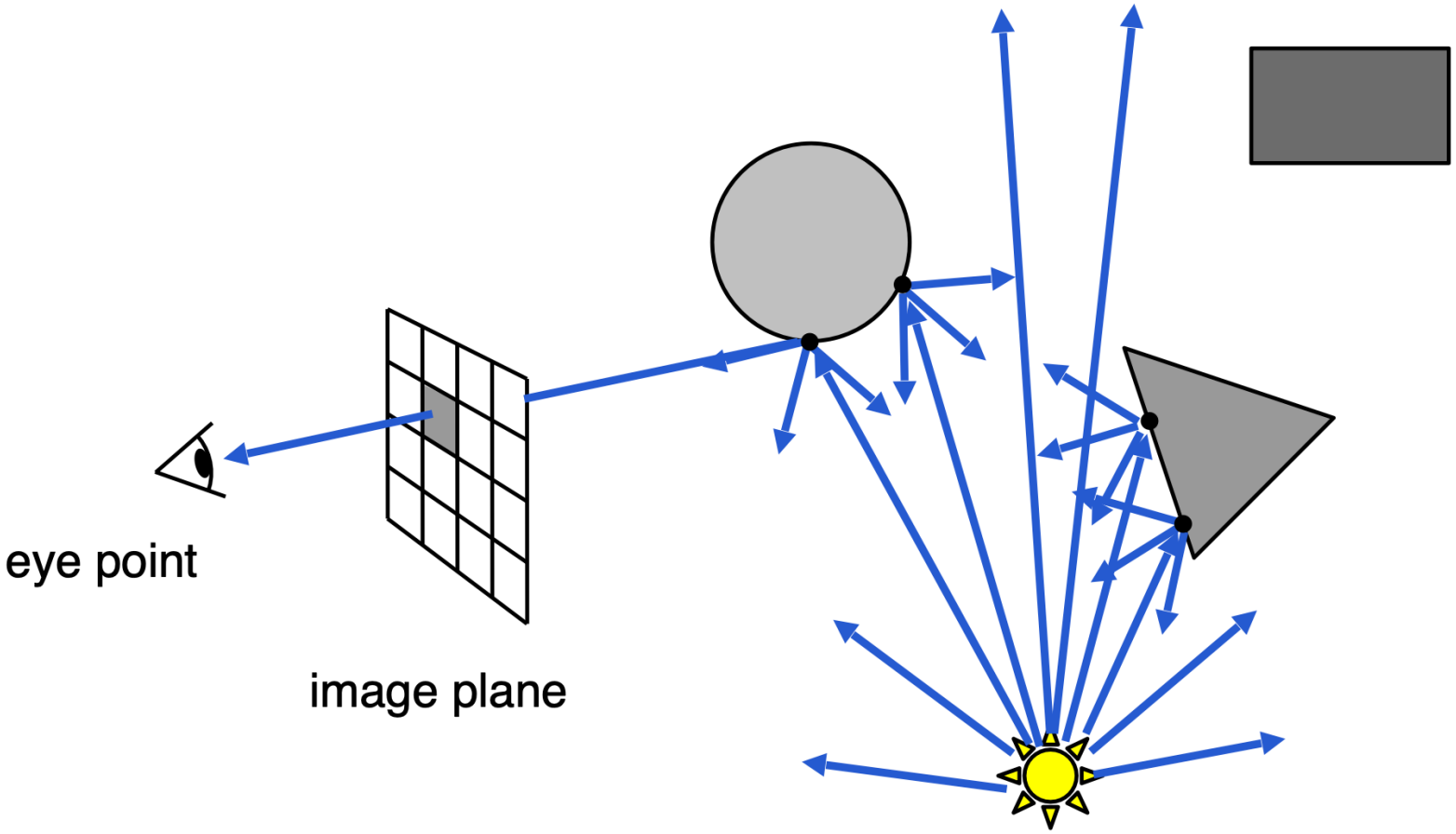
3D Scene Representation

?

Rendered Scene

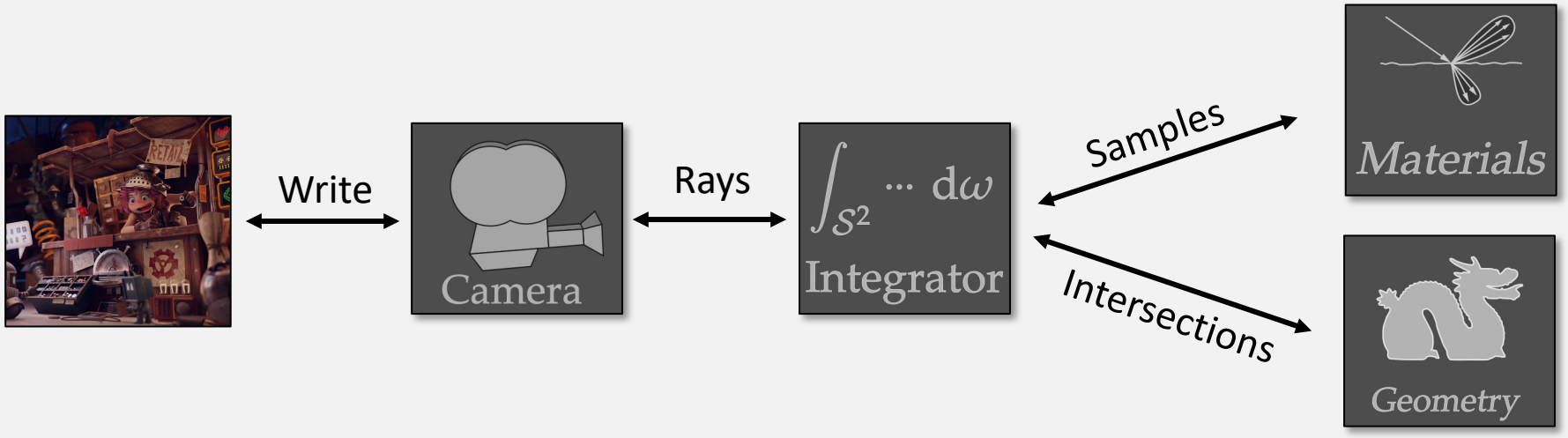
Rendering a scene

Light to Camera vs Camera to Light (Helmholtz Reciprocity)



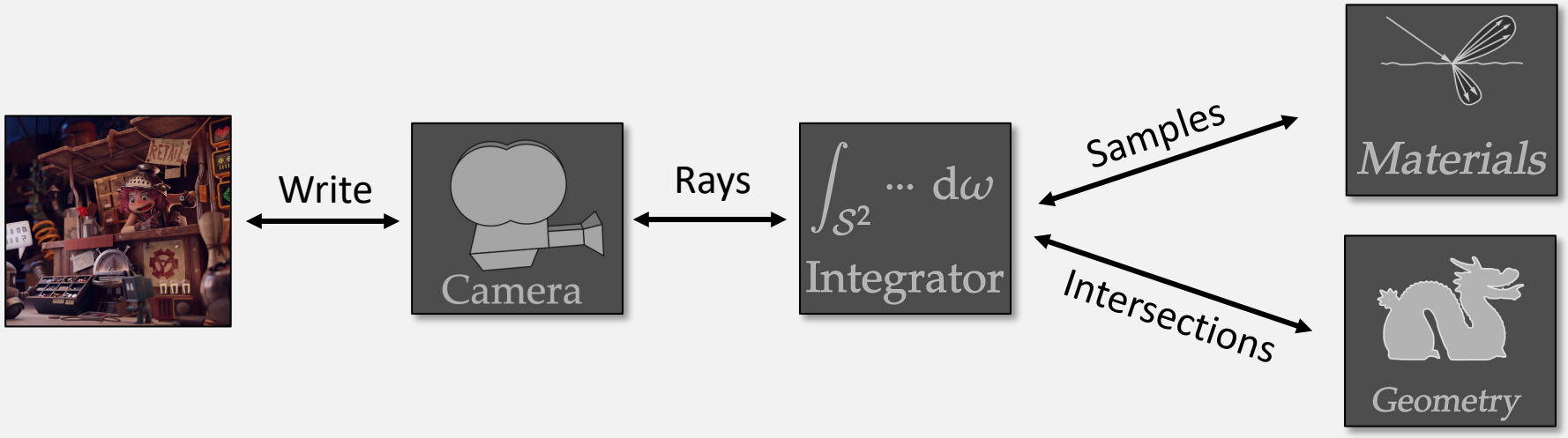
Rendering a scene

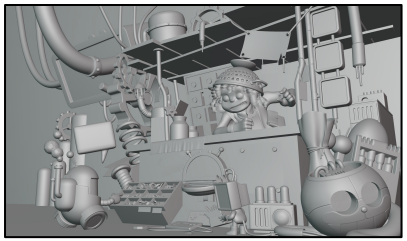
Rendering Engine *(What you will build!)*



Rendering a scene

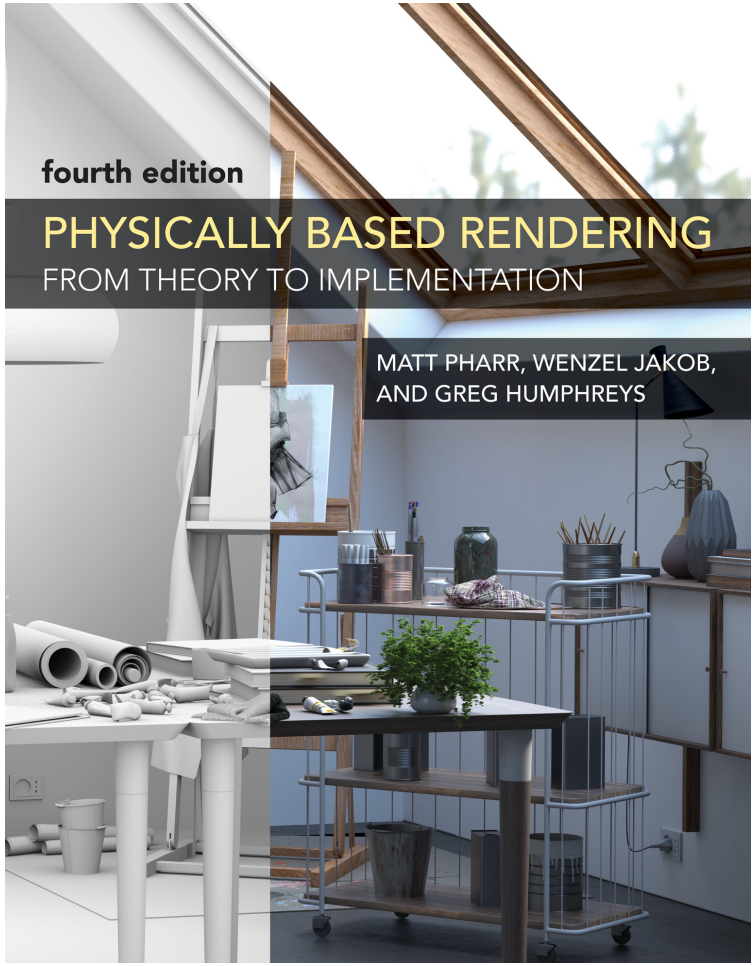
Rendering Engine *(What you will build!)*





Light Transport Simulation

PBRT



3rd edition freely available online!

Figure 11.13: Objects filled with participating media rendered with (left) strong backward scattering ($g = -0.7$) and (right) strong forward scattering ($g = 0.7$). Because the light source is behind the object with respect to the viewer, forward scattering leads to more light reaching the camera in this case.

HenyeyGreenstein provides a PhaseFunction implementation of the Henyey–Greenstein model.

```
<<HenyeyGreenstein Declarations>>=
class HenyeyGreenstein : public PhaseFunction {
public:
    <<HenyeyGreenstein Public Methods>>
private:
    const Float g;
};

<<HenyeyGreenstein Public Methods>>=
HenyeyGreenstein(Float g) : g(g) { }

<<HenyeyGreenstein Method Definitions>>=
Float HenyeyGreenstein::p(const Vector3f &wo, const Vector3f &wi) const {
    return PhaseHG(Dot(wo, wi), g);
}
```

The asymmetry parameter g in the Henyey–Greenstein model has a precise meaning. It is the average value of the product of the phase function being approximated and the cosine of the angle between ω' and ω. Given an arbitrary phase function p, the value of g can be computed as†

$$g = \int_{S^2} p(-\omega, \omega')(\omega \cdot \omega') \, d\omega' = 2\pi \int_0^\pi p(-\cos\theta) \cos\theta \sin\theta \, d\theta. \tag{11.5}$$

Thus, an isotropic phase function gives $g = 0$, as expected.

Any number of phase functions can satisfy this equation; the g value alone is not enough to uniquely describe a scattering distribution. Nevertheless, the convenience of being able to easily convert a complex scattering distribution into a simple parameterized model is often more important than this potential loss in accuracy.

Questions?