

PBRT: Appendix B

Saturday, February 6, 2021 8:53 PM

B Scene description interface

One API, two implementations: a set of functions to call and text input file ([format](#)). Both can be used to describe the scene.

Historically, there have been 2 approaches to scene description:

- The programmer configures the rendering pipeline at a low level; the API exposes internal algorithms and structures; the programmer tells the renderer *how* to render the scene. This is how interactive graphics APIs like Vulkan and Direct3D are used. For example, if reflections are wanted, one doesn't just mark or configure a surface as a mirror; instead, one does the work computing an environment map, writing to a texture, and then use it to shade the surface.
- The programmer specifies what the scene's objects, lights, and material properties are, at a high level of abstraction. This is the approach preferred by high-quality offline renderers.

Immediate mode or retained mode APIs:

- In immediate mode, the renderer receives a stream of commands that it processes as they arrive. The scene is rendered in real-time. The next frame reprocesses the entire stream again (which may contain changes). High-performance interactive graphics operate like this.
- In retained mode, the entire scene is first described and stored in memory and then rendering begins. The next frame reuses the scene description (with possible changes), because it remain in memory

PBRT uses immediate mode semantics: it reprocesses all the scene commands again for the next frame.

The PBRT API is designed with extensibility in mind: it's easy to add new Shapes, Integrators, Cameras, etc.

B.1 Parameter sets

The API doesn't know the exact parameters of, say, a given Shape or a given Camera.

The ParamSet class is a generic bundle of parameters. For the API (most API calls receive a ParamSet), the parameters don't have a meaning.

B.1.22 Adding to the parameter set

B.1.3 Looking up values in the parameter set

Functions that retrieve the value of a parameter from a parameter set by name also receive a default value, which is returned when a parameter with the given name is not found:

params.FindOneFloat("radius", 1. f).