RtRayGeometry

Introduction

A `RtRayGeometry` represents a geometric ray. This is how rays are provided to integrators (through `RixIntegrator::IntegrateRays()`) and how integrators provide rays to be traced to `RixIntegrator::GetNearestHits()` and `RixIntegrator::GetTransmission()`.

Fields

A `RtRayGeometry` contains several fields that need to be set by the integrator before it is given to the methods above. Some of these fields have straightforward interpretation and can be set to any desired value, while some other fields are expected to be set consistently with the current integrator and shading contexts.

- `RtPoint3 origin`: Ray origin
- `float originRadius`: Radius of the ray at origin. For ray emitted from pinhole cameras this should be zero, but when using an orthographic camera, it should be non-zero and depend on the 'size' of a pixel in the world.
- `RtVector3 direction`: Ray direction. This should be normalized.
- `float raySpread`: How much ray radius changes for each unit the ray travels. Supports values from 0 to 1 (1 corresponding to an angle of 45 degrees). The radius at a given hit can be used directly as texture/environment radius. or needs to be multiplied by 2 to be used as 'footprint' or as texture/environment filter diameters. `raySpread` is expected to be non-zero for rays emitted from a pinhole camera, and zero when using an orthographic camera.
- `float minDist`: raytracing will ignore hits closer than `minDist`. Only used for primary rays for now.
- `float maxDist`: raytracing will ignore hits further than `maxDist`.
- `float time`: Shutter-normalized time. Values are expected to be between 0 and 1.
- `uint16_t raytype`: Ray type. This will affect which objects can be hit by the ray.
- `RixBXLobeSampled lobeSampled`: Characteristics of the BxDF lobe this ray samples. This is mainly used for proper LPE behavior, but some parts of the renderer may also behave differently based on this.
- `RixShadingContext::Id shadingCtxId`: Shading context unique identifer. This refers to the shading context containing the shading point that triggered the creation of this ray.
- `int16_t integratorCtxIndex`: Index of the primary ray associated with this ray.
- `uint16_t shadingCtxIndex`: Index of the shading point that triggered the creation of this ray.
- `uint16_t rayId`: User id associated with the ray. This id value will be stored in the `RixShadingContext::rayId` field for hits associated with this ray.
- `RixLPEState* lpeState`: A pointer to the LPE state associated with the ray.

The following fields are used by the renderer internals and shouldn't be manually set:

- `uint32_t private0[5]`: Data reserved for internal use.
- `float opacityThreshold`: Opacity threshold associated with the ray. Used for better-stratified presence.
- `mutable uint16_t hitId`: Internal hit id.
- `uint16_t flags`: Internal flags.
- `uint16_t wavelength`: Ray wavelength. Can be retrieved by using `RixShadingContext::GetBuiltinVar(k_wavelength, ...)`.