Embree: High Performance Ray Tracing Kernels 2.3

Table of Contents:

1 Embree: High Performance Ray Tracing Kernels 2.3

1.1 Embree Tutorials

Embree comes with a set of tutorials aimed at helping users understand how embree can be used and extended. All tutorials exist in an ISPC and C version to demonstrate the two versions of the API. Look for files names

`tutorialXX_device.ispc`

for the ISPC implementation of the tutorial, and files named

`tutorialXX_device.cpp`

for the single ray C version of the tutorial. To start the C++ version use the

`tutorialXX`

executables, to start the ISPC version use the

`tutorialXX_ispc`

executables. You can select an initial camera using the -vp (camera position), -vi (camera lookat point), -vu (camera up vector), and -fov (vertical field of view) command line parameters:

```
./tutorial00 -vp 10 10 10 -vi 0 0 0
```

You can select the initial windows size using the -size command line parameter, or start the tutorials in fullscreen using the -fullscreen parameter:

```
./tutorial00 -size 1024 1024
./tutorial00 -fullscreen
```

Implementation specific parameters can be passed to the ray tracing core through the -rtcore command line parameter, e.g.:

```
./tutorial00 -rtcore verbose=2,threads=1,accel=bvh4.triangle1
```

The navigation in the interactive display mode follows the camera orbit model, where the camera revolves around the current center of interest. With the left mouse button you can rotate around the center of interest (the point initially set with -vi). Holding Control pressed while klicking the left mouse button rotates the camera around its location. You can also use the arrow keys for navigation.

You can use the following keys:

F1
: Default shading

F2
: Gray EyeLight shading

F3
: Ambient occlusion shading

F4
: UV Coordinate visualization

F5
: Geometry normal visualization

F6
: Geometry ID visualization

F7
: Geometry ID and Primitive ID visualization

F8
: Simple shading with 16 rays per pixel for benchmarking.

F9
: Switches to render cost visualization. Pressing again reduces brightness.

F10
: Switches to render cost visualization. Pressing again increases brightness.

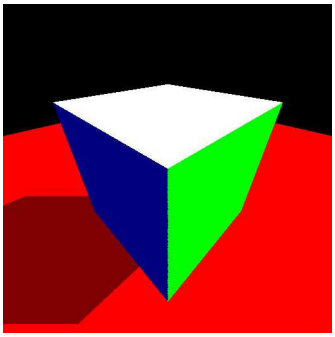
f
: Enters or leaves full screen mode.

c
: Prints camera parameters.

ESC
: Exists the tutorial.

q
: Exists the tutorial.

1.1.1 Tutorial00



This tutorial demonstrates the creation of a static cube and ground plane using triangle meshes. It also demonstrates the use of the `rtcIntersect` and `rtcOccluded` functions to render primary visibility and hard shadows. The cube sides are colored based on the ID of the hit primitive.