minrt

2015-03-10

Daniel Pohl

minrt is a minimal ray tracer that interacts with minclient for in-home streaming. It is not meant as a fully featured software, but as a reference implementation. Therefore, important features like texturing or image loading are not included.

The software comes **without official support**. But you could try to reach some of the team members, e.g. the author of this document with firstname.lastname@intel.com.

The main target for this software is Windows 8.x with Visual Studio 2013, Update 4. Some internal features (e.g. CILK) work only together with Intel-specific tools like Intel Parallel Studio 2015 Update 2. But it is not required to be installed for minrt to be working.

The software should also run under MacOS X and Linux.

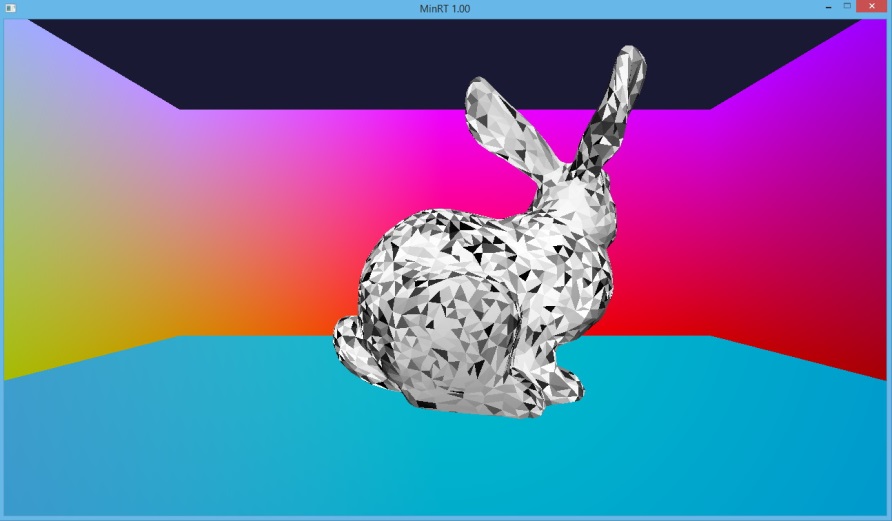
# Test 1: Getting the ray tracer to work

After successful compilation it is best to test the ray tracer without in-home streaming. To run on a single machine, without any networking, make sure that these values are set in Engine.cpp.

bool Engine::server = false;

bool Engine::dedicated = false;

If your machine has only support for SSE (4-wide SIMD), but not AVX (8-wide SIMD), make sure to set Engine::embreeIntersect = 4. For AVX the value should 8. Other values are currently not used. When launching the executable an image like this should appear.



Using W, A, S, D, space (up) and c (down) the scene can be navigated. Press ESC to exit. The frame rate is displayed in the console window.

Now change the server and dedicated values back to the server mode.

bool Engine::server = true;

bool Engine::dedicated = true;

More tests can be found in the minclient project.

# Other information

The screen resolution can be set with those variables.

int Engine::screenWidthRT = 1280;

int Engine::screenHeightRT = 720;

int Engine::screenWidthGL = 1280;

int Engine::screenHeightGL = 720;

The minclient needs to have exactly the same values.

If bool Engine::debugLatency = true then every alternating frame a different color will be added to the frame buffer. For measuring motion-to-photons latency with a high speed camera this can be helpful to clearly differentiate between the frames.

The default port for networking is int Engine::serverPort = 2000. It can be changed over the command line with e.g. -port 2001. This is helpful, if multiple minrt servers are running.