

Embree 2.17.2

Installation and verification

Luís Paulo Santos, February, 2018

Throughout this course we will use Embree as the ray tracing engine. Embree is a collection of optimized ray tracing kernels, developed by Intel, with support for vector processing such as SSE, AVX, AVX2 and AVX-512. Embree selects at runtime ray traversal and BVH build algorithms that best match the instruction set of your CPU.

This tutorial is just a brief summary of the extended information you can find at:

<https://embree.github.io/>

Installation

During the semester we will be using **Embree 2.17.2**, since version 3.0.0 is still a beta release.

Follow instructions available at <https://embree.github.io/downloads.html> for your operating system.

Even though pre-built versions of the library can be found and used I suggest you **build Embree from the sources**. We will need to have a look and eventually modify the source code of the tutorials, and this is one way of getting access to it. The sources can be **downloaded** from one of the links below:

- <https://github.com/embree/embree/archive/v2.17.2.zip>
- <https://github.com/embree/embree/archive/v2.17.2.tar.gz>

Extract Embree sources to a folder where you have full access permissions. We will refer to this folder as \$EMBREE_SOURCES\$.

NOTE: **DO NOT use special/punctuation characters in any folder/files names**. This is NOT recommended in ANY circumstance

You will need Intel Thread Building Blocks (TBB). It might either be available on your system or you might have to download it from <https://www.threadingbuildingblocks.org/download>

We suggest you **extract** Intel Thread Building Blocks (TBB) to a folder within \$EMBREE_SOURCES\$.

Also **install** CMake (<http://www.cmake.org/download/> - version 2.8.11 or higher).

NOTE: in order to build Embree you need to have a C++ compiler installed in your system. Obviously, there are many different compiler options available and these depend on the particular operating system you are using.

For **Windows** operating system and **Visual Studio C++** an option is outlined below. Remember that as a student of a high education institution you might be entitled to download Visual Studio from other sources:

1. If Visual Studio is not available in your system, download Visual Studio Community 2017 <https://www.visualstudio.com/downloads> (after 30 days, login may be required to stay free-to-use)
 2. If Visual Studio installer did not prompt to install Visual C++ compilers, use <https://www.visualstudio.com/downloads/#build-tools-for-visual-studio-2017>
-

3. According to Windows version, download Windows SDK: <https://developer.microsoft.com/en-us/windows/downloads>
-

CMake

Open the CMake graphical user interface.

Press the “Browse source...” button and select the folder where Embree has been extracted (and referred to herein as \$EMBREE_SOURCES\$).

Press the “Browse build...” button and select the folder where you want Embree to be built (a good idea might be \$EMBREE_SOURCES\$/build).

Press **”Configure”**. Note that you have to indicate **which project generator** to use. Very **carefully verify** that the selected generator corresponds to the version of the compiler you will be using and also to the appropriate operating system (e.g., Win64 *versus* Win32).

The CMake configurator will start working and you might have a set of errors. These have to be handled one by one. A few examples follow:

- if you have opted to not use Intel ISPC then you have to uncheck “EMBREE_ISPC_SUPPORT” in CMake graphical user interface;
- I suggest you change the “CMAKE_CONFIGURATION_TYPES” to “Release;” deleting the other options;
- Change the “CMAKE_CONFIGURATION_TYPES” to the root folder of your TBB installation. If you followed the suggestion above this will be a subfolder in \$EMBREE_SOURCES\$.

Keep pressing **”Configure”** in CMake and making changes in CMake variables until it succeeds. Then press **”Generate”**. It should succeed without a problem. You can now close CMake and are ready to build Embree.

Building Embree

After configuring and generating with CMake you should be able to build Embree using your compiler and, eventually, associated IDE.

This building process depends a lot on the operating system you are using. All the required information has been assembled by CMake in the \$EMBREE_SOURCES\$/build folder.

If your operating system is Linux or MacOS there will probably be a Makefile and you just have to call the make utility from this folder, eventually using a *shell*.

If your operating system is Windows and you are using Visual Studio then there will be a embree2.sln file. Double clicking on it will open Visual Studio. Before starting the build make sure within Visual Studio that the appropriate configuration is selected, i.e., the “Release” configuration and most probably, but depending on your system, the x64 architecture. Now start the build pressing F7.

Embree has a large code support basis. Building will take a while, how long depends on your particular system. Be patient.

Verification

After the build you should find all the Embree libraries and tutorials applications on your selected target directory, which is \$EMBREE_SOURCES\$/build/Release. Note that this folder contains

the dynamic and shared objects (e.g. .dll and .lib under Windows), as well as the tutorial executables.

Before executing these files you need to grant access to tbb. There are several alternatives on how to do this, which depend on the target operating system.

On a Windows 64 bit system with Visual Studio 2015 version 14, locate the appropriate version of the tbb.dll file, which might be at \$EMBREE_SOURCES\$/TBB_EXTRACT\$/bin/intel64/vc14 You can copy the contents of this folder to \$EMBREE_SOURCES\$/build/Release, or else you might set the environment variable PATH to also point to the TBB folder.

Verify your installation running for instance the path tracer tutorial:

- make sure you identify the directory where the geometric models are made available (copy the models folder to the folder where the tutorials have been built, from the \$EMBREE_SOURCES/tutorials/models folder)
- execute the path tracer (you can change the loaded model by writing: -c <model path>) – check for instructions at <https://embree.github.io/tutorials.html>)
- Note that you can load other models made available at Intel Embree's page:
 - <https://github.com/embree/models/releases/download/release/crown.zip>
 - https://github.com/embree/models/releases/download/release/asian_dragon.zip