

IDIG4002 - Computer Graphics Fundamentals and Applications

Lab 1: Exercise 1.

1. PBRT-V3 compilation

Windows

1. Install CMAKE for windows
 - a. Go to <https://cmake.org/download/> and download the appropriate version for your computer.
 - b. Install the binary release
2. Install git for windows: distributed version control system
 - a. Go to <https://www.git-scm.com/downloads> and download for windows.
 - b. Install the binary release
3. Install visual studio 2019
 - a. Go to <https://visualstudio.microsoft.com/downloads/> and download and install community version for windows.
4. Get pbrt-v3 from the official git repository
 - a. Go to <https://github.com/mmp/pbrt-v3>.
 - b. Copy the Clone with HTTPS link
 - c. Open windows command line or git window
 - d. Change your current directory to your development folder

```
$ cd YOUR_PATH
```
 - e. Clone pbrt-v3 from github

```
$ git clone --recursive REPOSITORY_LINK
```
5. Compile pbrt-v3
 - a. Go in to the cloned pbrt-v3 folder and create a directory to build the source code.

```
$ cd pbrt-v3
```

```
$ mkdir build
```
 - b. Open and use CMAKE GUI to build the source code
 - i. Select the source code and build directories, then click configure.
 - ii. For the project generator select the version of your visual studio – visual studio 17
 - iii. Platform for generator should be x64 for 64 bit development. Otherwise it will build the 32 bit version.
 - iv. Click finish.
 - v. When the configuration is done, click generate to generate the solution files.

- c. Go to your build directory and open the generated PBRT-V3.sln file with visual studio 19.
 - d. To build the solution, use release solution configuration and x64 for solution platform. Finally click on build solution from build menu – or press F7.
 - e. When the build is done, pbrt and other related executable files will be generated in YOUR_PATH/build/Release folder.
6. Testing pbrt-v3 for rendering
 - a. Run pbrt explicitly from the command line by typing the path to where the pbrt binary is located. For example
\$./YOUR_PATH/build/Release/pbrt scenes/ killeroo-simple.pbrt
 - b. It is also possible to add the path to the pbrt executable to the execution environment variable, which allows you to run the binary without the path.
\$ pbrt scenes/ killeroo-simple.pbrt

Linux

1. Open terminal and Install cmake, git and gcc


```
$ sudo apt upgrade
$ sudo apt update
$ sudo snap install cmake -classic
$ cmake --version
$ apt-get install git
$ git --version
$ sudo apt install build-essential
$ sudo apt-get install manpages-dev
$ gcc --version
```
2. Change the current directory to your development folder.


```
$ cd YOUR_PATH
```
3. Get pbrt-v3 from the official git repository
 - a. Go to <https://github.com/mmp/pbrt-v3>.
 - b. Copy the Clone with HTTPS link
 - c. Clone pbrt-v3 from github
\$ git clone --recursive REPOSITORY_LINK
4. Compile pbrt-v3
 - a. Go in to the pbrt-v3 folder and create a new directory for building the source code.
\$ cd pbrt-v3
\$ mkdir build
\$ cd build
 - b. Build the source code with cmake
\$ cmake ../
\$ make
 - c. When the build process is finished, all the necessary executables will be found in the build directory. List the files in the folder.
5. Testing pbrt-v3 for rendering

- a. Run pbrt explicitly from the command line by typing the path to where the pbrt executable is located. For example
\$./YOUR_PATH/build/Release/pbrt scenes/ killeroo-simple.pbrt
 - b. It is also possible to run the binary without the path.
 - i. Add **export** PATH=\$YOUR_PATH/build/Release/pbrt:\$PATH to .bash_profile file.
\$ pbrt scenes/ killeroo-simple.pbrt
2. PBRT rendering output: After pbrt is done rendering, it will print statistics of the rendering process to the screen and output image in EXR format (killeroo-simple.exr). The EXR image stores linear radiance values, which can be rendered fully with HDR displays. However, most screens can only display 8 bits of color information per RGB channel (SDR displays). Therefore, the HDR values must be properly tone mapped for SDR output. Different software and HDR libraries such as Adobe Photoshop, GIMP 2.9.2+, or OpenEXR's exrdisplay, Matlab HDRToolBox, and related others can be used to directly display EXR images. Pbrt-v3 also comes with image conversion tool called imgtool. Use imgtool to convert the exr image to png image.

```
$ imgtool covert killeroo-simple.exr killeroo-simple.png
```