PBRTv4-系列1-初步了解与代码的编译

Dezeming Family

2023年8月5日

DezemingFamily系列文章和电子文档**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了DezemingFamily的系列文章，可以从我们的网站[https://dezeming.top/]找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

目录

一　PBRTv4的新特性

也该开始撰写一下PBRTv4[2]的相关文章了，虽然当初并不太看好PBRT的后续版本（主要是越来越不适合初学者），但对于有一定基础的研究者来说，PBRTv4显然更有诱惑力：支持GPU渲染，以及支持一些更前沿的学术领域技术。

最近一直在设计自己的渲染器方面有些吃力，主要还是个人能力有限，且考虑是否要依赖于Optix等现有工具，诚然即使使用了Optix，除了包围盒以外的其他资源还是都得需要自己去创建。在撰写论文或者做实验时，一个比较好的工具是很有帮助的。Mitsuba 0.5虽然代码和功能都很不错，但可用的复杂场景不太够，而高版本的Mitsuba在编译和开发也都面临了诸多困难。虽然我开启了一个系列去介绍重新组织和编译Mitsuba v1，但仍不太建议使用它作为科研工具。

我们在介绍本系列文章时，希望大家能有PBRTv3的一些基础，大概就是《PBRT3-基本理论与代码实战》的前几本小册子即可。

1 1　PBRT的新改进

渲染都是在光谱上进行的（采样光谱，将光谱离散化），RGB仅仅作为场景描述和纹理映射。

支持基于null-scattering的VolPathIntegrator，较紧的多数(Tighter majorants)用于通过单独的低分辨率多数网格与GridDensityMedia进行零散射。现在支持发射体和具有RGB值吸收和散射系数的体。

GPU路径提供了基于CPU的VolPathIntegrator的所有功能，包括体散射、次表面散射、pbrt的所有相机、采样器、形状、灯光、材料和BxDF等，性能大大快于在CPU上渲染。

实现了更多种表面散射材质。

实现了多种光采样方法，比如”Many-light” sampling/Solid angle sampling等。

对采样器类进行了各种改进，包括更好的随机化和实现Ahmed和Wonka的蓝噪声Sobol采样器的新采样器。

一种新的GBufferFilm现在可以提供每个像素的位置、法线、反照率等。（这对于去噪和ML训练特别有用。）

路径正则化（可选）。

添加了双线性patch primitive（Reshetov 2019）。

ray–shape求交精度的各种改进。

大多数低级采样代码都被分解为独立的函数，以便于重用。此外，还提供了反转(invert)许多采样技术的功能。

单元测试覆盖范围已大幅增加。

二　下载和安装PBRTv4

安装个Git Bash，然后用以下命令下载即可：

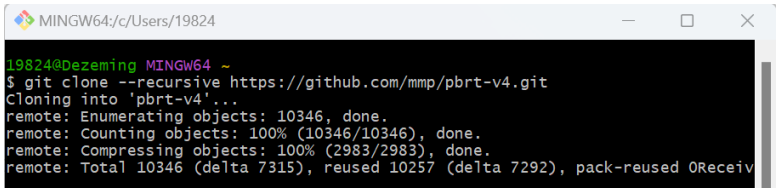
```
git clone --recursive https://github.com/mmp/pbrt-v4.git
```

recursive参数表示需要clone依赖库，如果之前拷贝的版本没有依赖，就调用下面的命令来生成依赖：

```
git submodule update --init --recursive
```

如果clone失败，需要重新clone，就先调用下面的命令来清理，然后重新clone：

```
rm -rf pbrt-v4
```



默认的git下载目录在（YourName是你的用户名）：

```
C:/Users/YourName
```

拷贝到你自己建立的目录下即可。

三　编译基于GPU的PBRTv4

为了在GPU上运行，PBRT有以下要求。这些要求有效地使在对核心系统进行有限更改的情况下将pbrt引入GPU成为可能：

- GPU上的C++17支持，包括使用C++ lambdas启动内核。
- Unified memory，使CPU可以为GPU上运行的代码分配和初始化数据结构。
- 用于GPU上ray-object求交的API。

实际上，这些功能目前只能通过CUDA和OptiX在NVIDIA GPU上使用。pbrt的GPU方法当前需要CUDA 11.0或更高版本以及OptiX 7.1或更高版本。

3 1　安装Visual Studio 2019

注意PBRT只能用Visual Studio 2019来编译，所以我们只能安装VS2019。安装方法比较简单，具体请可以参考网上资料，这里不再赘述。

3 2　安装CUDA 11或12

注意这里需要跟你的显卡计算能力等相匹配。我的GPU因为是很新的RTX4090，所以我安装了CUDA 12才最终编译成功。老点的RTX显卡安装CUDA11.3以上版本应该没有太大问题。

安装方法比较简单，具体请可以参考网上资料，这里不再赘述。注意你的计算能力

3 3　安装Optix 7.5

安装Optix 7.5也很简单，在[3]中下载Optix 7.5，注意虽然标注的是win10，但是win11系统安装和使用也是没有什么问题的。

3 4　编译PBRT v4

PBRT的CMAKE构建脚本会自动尝试在通常的地方查找CUDA编译器；cmake输出将指示它是否成功。

对于Optix依赖，有必要手动设置cmake PBRT_OPTIX7_PATH配置选项以指向OptiX安装，比如Optix默认安装路径是：

```
C:\ProgramData\NVIDIA Corporation\OptiX SDK 7.5.0
```

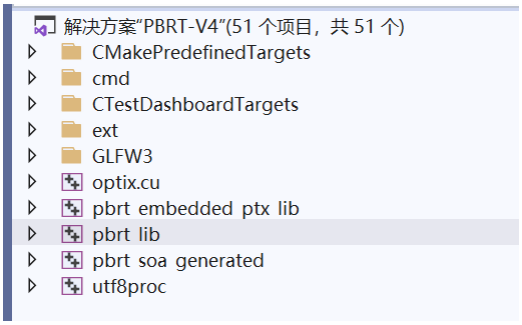
默认情况下，pbrt目标的GPU着色器模型是基于系统中的GPU自动设置的。或者，PBRT_GPU_SHADER_MODEL选项可以手动设置（例如-DPBRT_GPU SHADER_MODEL=sm_80）：

注意如果你的CUDA版本或者GPU型号等出现不兼容问题，就有可能出现编译错误：

```
nvcc fatal : Unsupported gpu architecture 'compute_89'
```

这个时候就需要重新安装其他版本的CUDA然后再试，我曾尝试试图修改CMakeLists.txt文件的设置来解决问题，但都没有很好地解决。

即使在使用GPU支持进行编译时，pbrt也默认使用CPU，除非提供–GPU命令行选项。请注意，在使用GPU进行渲染时，–spp命令行标志有助于轻松增加每像素的采样数。此外，使用tev观看渲染进度也非常有趣。



注意默认是编译为Release模式，除非你指定DCMAKE_BUILD_TYPE=Debug，才能编译为Debug模式。

四　PBRTv4的场景的使用

PBRT的场景资源见[4]，下载以后放在自己的数据文件夹里，比如我们放在了：

```
D:\DataSets\PBRTv4-Scenes\
```

随便解压一个场景，里面的.pbrt文件，该文件就是场景文件。在PBRT v3的系列文章中我们已经详细介绍过场景的parse，文件格式基本上是没有什么变化的，也比较容易理解。在源码的pbrt_lib项目中，可以看到parser.cpp文件，用以加载场景文件。

我们去编译好的Release版本目录下查看，里面有pbrt.exe：

```
D:\Develop\TheThridPartyCode\pbrt-v4-Build\Release
```

cmd进入该目录，运行的方法是：

```
pbrt.exe D:\DataSets\PBRTv4-Scenes\living-room\scene-v4.pbrt
```

渲染结果保存在pbrt.exe所在目录下。

若要使用GPU渲染，需要在命令行加上 –gpu 命令，此时，场景文件指定的渲染积分器是无效的，因为GPU只支持VolPathIntegrator积分器的实现。–interactive命令可以变成可交互渲染，但是要注意，一旦渲染的spp数完成，就会关闭，因此，最好在文件中把spp数设置的大一点。完整的命令如下：

```
pbrt.exe --gpu --interactive D:\DataSets\PBRTv4-Scenes\living-room\scene-v4.
    pbrt
```

此时就会弹出一个窗口来显示渲染的内容。

作为pbrt的一部分构建的imgtool程序在GPU构建中提供了对OptiX去噪器的支持。去噪器能够仅对RGB图像进行操作，但对包括反照率和法线等辅助通道的“深度”图像可以获得更好的结果。渲染时将场景的“Film”类型设置为“gbuffer”，并对图像格式使用EXR会导致pbrt生成这样的“深度”图像。在任何一种情况下，使用去噪器都很简单：

```
imgtool denoise-optix noisy.exr --outfile denoised.exr
```

参考文献

[1] https://github.com/mmp/pbrt-v4

[2] https://pbrt.org/

[3] https://developer.nvidia.com/designworks/optix/downloads/legacy

[4] https://pbrt.org/resources