

初识 Tungsten 渲染器

Dezeming Family

2023 年 7 月 7 日

DezemingFamily 系列文章和电子书**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 的系列电子书，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 基本介绍	1
1 1 使用方法	1
1 2 编译源代码	1
二 代码结构	2
2 1 基本代码结构	2
2 2 实用工具	2
参考文献	2

一 基本介绍

Tungsten 是 2014 年初为 ETH 的图像综合渲染比赛编写的基于物理的渲染器，它上面显示的图像获得了第一名。它旨在用于图形研究中的离线渲染，不过截止到目前已经五年没有再开发过了。它在 Github 的地址为 [3]。

Tungsten 使用 C++11 编写，并利用 Intel 的 Embree 光线追踪库进行快速的光线-物体相交测试。它在性能方面进行了优化，并利用多线程和 SIMD 技术。在低级优化的基础上，复杂的积分器、多重重要性采样和自适应立体角光源选择 (solid-angle-adaptive light selection) 减少了整体运行时间，以消除噪声，自适应采样确保工作集中在图像中最嘈杂的区域。Tungsten 支持许多几何形状和材质模型，使场景更有趣，还支持参与介质，包括能够渲染太空中行星的大气介质。

在当前的状态下，Tungsten 更多地是基于学术兴趣而不是实际使用。关键功能，如动态模糊，还不存在，而缺乏文档使得除了原作者之外的其他人很难使用这个渲染器，这些也是正在努力解决的问题，但由于持续缺乏空闲时间，进展往往很慢。

1.1 使用方法

我们从 [4] 中下载工程（可执行程序）和代码。然后从 [2] 中挑一些场景来下载。之后，先解压工程压缩包，看到里面有 tungsten.exe，我在工程目录下新建了一个文件夹，叫 Examples，用来存放场景文件。

运行方法是：

```
1 tungsten ./Examples/bathroom/scene.json
```

之后启动渲染：

```
1 D:\Develop\C++ Test\tungsten-0.2.2>tungsten ./Examples/bathroom/scene.json
2 Loading scene './Examples/bathroom/scene.json' ...
3 Starting render ...
4 Completed 16/64 spp
5 Completed 32/64 spp
6 Completed 48/64 spp
7 Completed 64/64 spp
8 Finished render. Render time 1m 8s 948ms
```

最终渲染结果会被保存在场景文件的目录下。

1.2 编译源代码

直接用 CMake 编译就好，除了提前设置 Visual Studio 和 Qt5 的环境以外，不需要其他的额外配置。

我编译的环境是 Visual Studio 2015，以及 CMake 3.23.3，Qt 是 Qt5.7，在相似环境下基本没有问题。

在 CMake 中 Configure 两次，然后 Generate，再打开工程即可编译。注意编译的结果没法直接运行，因为大部分需要输入参数。此外还需要 data 文件夹，里面存放了代码运行时需要的数据：

```
1 editor
2 example-scenes
3 materialtest
4 mc-loader
5 shaders
```

二 代码结构

2.1 基本代码结构

src/core/ 包含了基元求交、材质、采样、积分器等所有代码。它是渲染器的强大部分，如果您有兴趣研究代码，它也是开始的地方。

src/thirdparty 包含项目中使用的的所有库。它们包含在 repository 中，因为它们中的大多数要么是微小的单个文件库，要么在 embree 的情况下，必须进行修改才能使用渲染器。

src/tungsten 包含渲染应用程序本身，它只是核心渲染代码的一个小命令行接口。

2.2 实用工具

除了核心渲染器，Tungsten 还提供了一些工具，使内容创建更容易。

tungsten_server 是渲染器的独立版本，带有内置的 HTTP 状态服务器。

它将提供以下文件：

/render: 当前帧缓冲区（可能处于不完整状态）。

/status: 一个 JSON 字符串，包含有关当前渲染状态的信息。

/log: 渲染日志的文本版本。

用下面代码可以获得更详细的信息：

```
1 tungsten_server --help
```

scenemanip 提供了一系列操作场景文件的工具，其中包括将场景和所有引用资源（纹理、网格等）打包到 zip 中的功能。

hdrmanip 将 HDR 图像转换为低动态范围。

```
1 hdrmanip -o output.png input.exr
```

obj2json 将解析 Wavefront OBJ srcFile.obj，包括对象层次结构和材质，并创建一个场景文件 dstFile.json，再现层次结构和材质分配。

```
1 obj2json srcFile.obj dstFile.json
```

json2xml 将解析场景文件 srcFile.json，并将其转换为与 Mitsuba 渲染器兼容的 XML 场景描述。

```
1 json2xml srcFile.json dstFile.xml
```

editor 是一个用 Qt 和 OpenGL 编写的极简主义场景编辑器。它支持相机设置、操纵变换、合成场景以及我觉得有用的其他一些功能。该代码既不特别干净也不稳定，因此使用该代码会存在风险。

参考文献

[1] <https://benedikt-bitterli.me/tungsten.html>

[2] <https://benedikt-bitterli.me/resources/>

[3] <https://github.com/tunabrain/tungsten>

[4] <https://github.com/tunabrain/tungsten/releases/tag/v0.2.2>