

Monte Carlo Path Tracer

梁潇 21721237

Introduction

这是一个采用蒙特卡洛路径追踪算法，冯模型来实现的光照渲染器。

Features

- 全部代码没有使用第三方库，使用的都是 C++ 的标准库
- 实现了类似 vector3f 的向量类
- 使用 OpenMP 进行并行加速
- 在实现 scene1 和 scene2 的时候，修改了 obj 文件，使用自定义的 sphere 类替代了原始的使用三角面片表达的球体，极大的减少了计算量

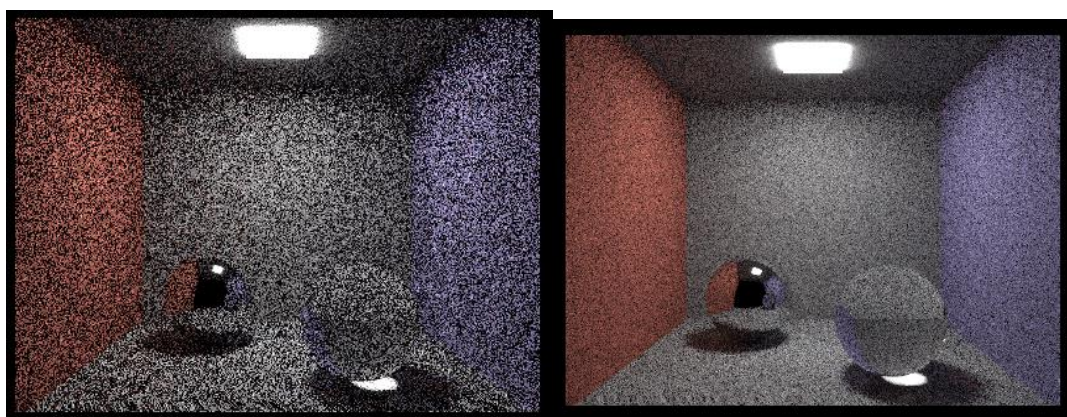
Demo

运行环境为：

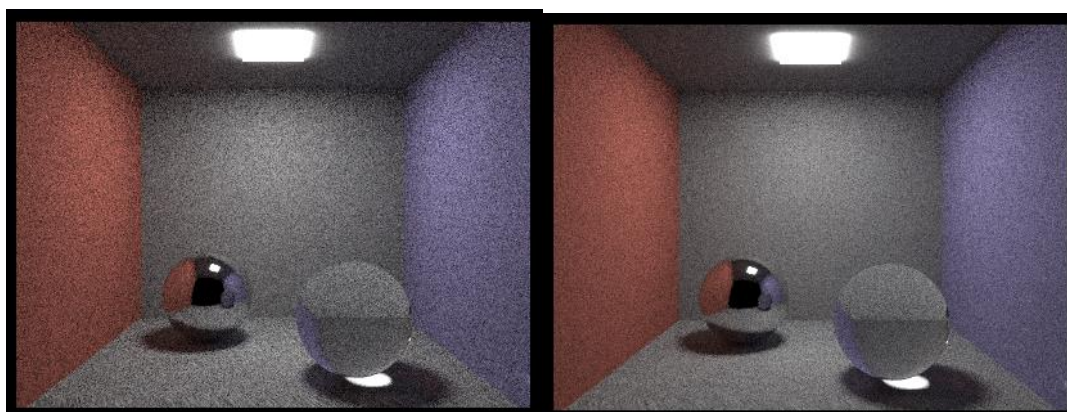
- CPU: i7-7700k, 主频 4.5GHz, 四核八线程
- OS: Windows 10
- Compiler: Visual Studio 2017 C++ Compiler

以下为运行结果：

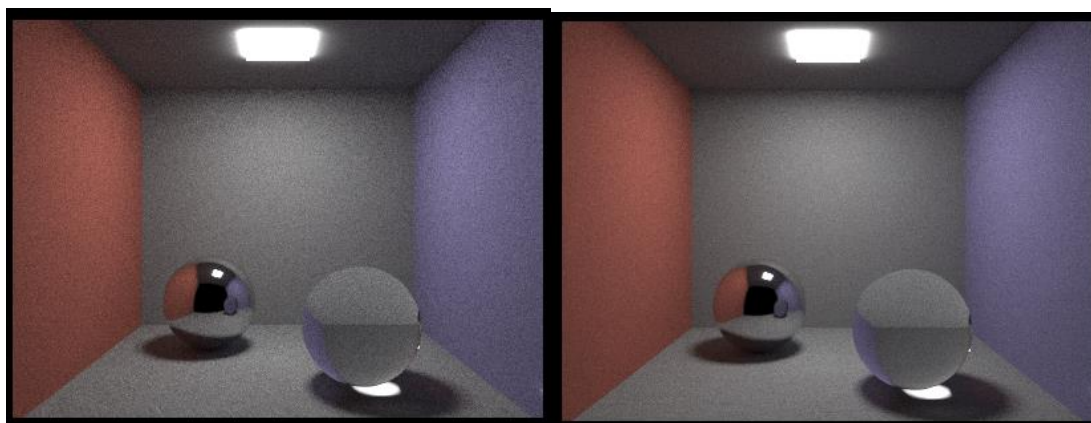
- Scene 1:



sample = 100 && 500



sample = 1000 && 2000



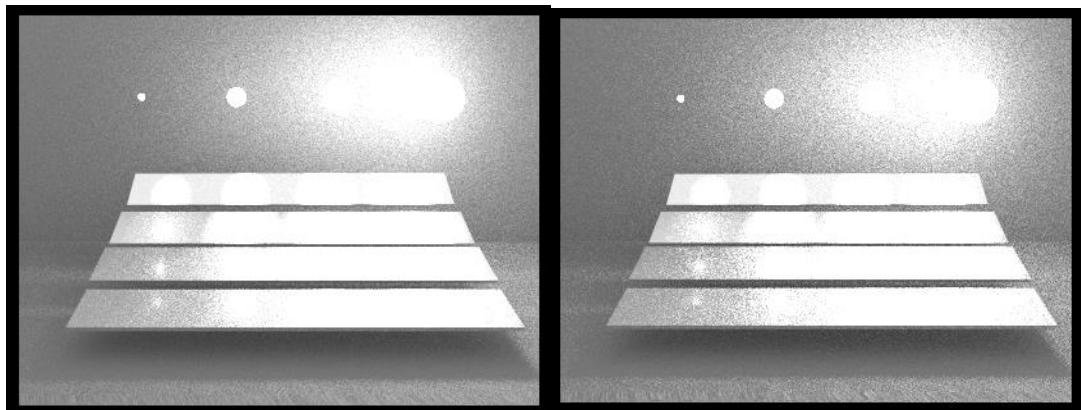
sample = 5000 && 10000

随着 sample 数量的增大，渲染结果的噪点也逐步变少。

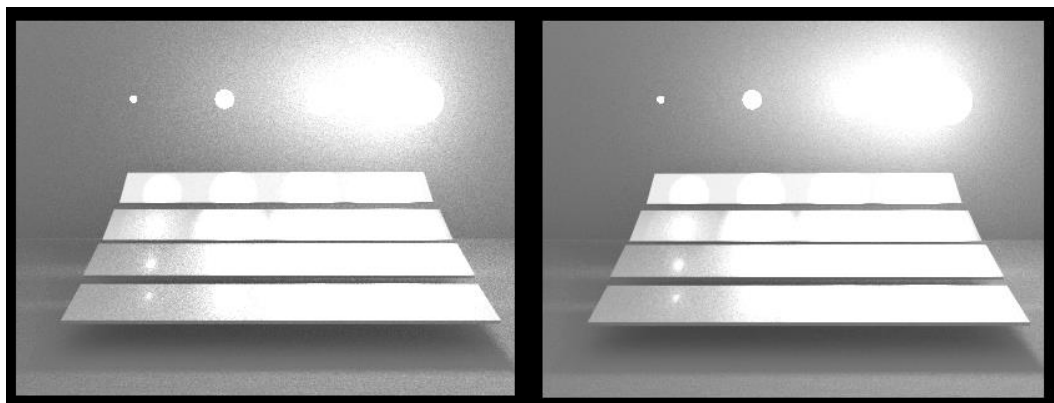
```
D:\VSproject\RayTracing\x64\Release\RayTracing.exe
mtl file name is scene01.mtl
Read from scene01.mtl success!
Material Number = 5
Read from scene01.obj success!
Vertex Number = 439
Object Number = 5
Start Tracing...
Rendering (10000 samples per pixel) 100.00%End Tracing, total time 8626.00 seconds...
请按任意键继续. . .
```

400×300 分辨率下，运行 10000 个 sample，最终耗时 8626 秒。

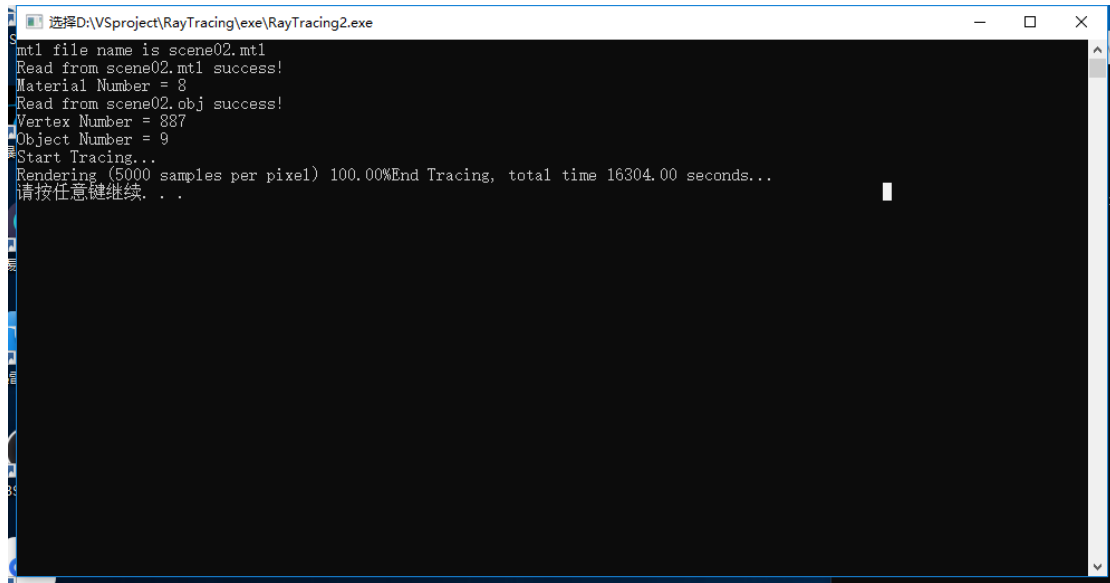
· Scene 2:



sample = 500 && 1000



sample = 2000 % 5000



```
选择D:\VSproject\RayTracing\exe\RayTracing2.exe
mtl file name is scene02.mtl
Read from scene02.mtl success!
Material Number = 8
Read from scene02.obj success!
Vertex Number = 887
Object Number = 9
Start Tracing...
Rendering (5000 samples per pixel) 100.00%End Tracing, total time 16304.00 seconds...
请按任意键继续. . .
```

400×300 分辨率下，运行 5000 个 sample，最终耗时 16304 秒。

Reference

<https://github.com/lllloadinglll/PathTracing>

<https://github.com/Aries-orz/Ari-RayTracing>