

黑洞渲染器

2025 年 6 月 7 日

1 简介

电影《星际穿越》中名为“卡冈图雅”的黑洞给了我很大的震撼，从第一次看到它我就想自己复现一遍。在 24 秋期末看到有一个别的团队做了一个效果不错的，这让我重新想起之前的计划，于是 25 春选了计算机图形学、开始自己看广义相对论，准备完成这个项目。感谢大作业给了我一个这样的机会（不然可能拖到暑假自己慢悠悠地做了）。

本项目主要使用 cpp 编写，采用 CMake 构建，调用库包含 Eigen、GLAD、GLFW、ImGui、GSL 以及 Matplotlibcpp。最后两个主要用于测试 (GSL 为了显示氢原子的轨道以测试体渲染，Matplotlibcpp 以测试测地线方程求解器)。以及部分 glsl 语言实现渲染等效果。项目也包含了部分 python 语言编写的符号计算程序 (用于计算任意时空下的联络以便计算测地线)。

本项目使用的环境是 VSCode,CMake 配置使用 Visual Studio Community 2022 Release - amd64. 如果您在运行 GEODESIC_TEST 或 GEODESIC_AUTO_TEST 项目，请确保您的环境中存在 python 以及 numpy.

2 项目内容

本项目最终要实现黑洞及其吸积盘的渲染。

2.1 测地线计算

在广义相对论下，时空是一个定义了度规的四维流形，物体不受其他作用时的运动轨迹是四维流形上的一条测地线（而光对应的测地线上的切矢量长度恒为 0）。测地线方程如下：

$$\frac{d^2x^\mu}{d\lambda^2} = \Gamma^\mu_{\nu\eta} \frac{dx^\nu}{d\lambda} \frac{dx^\eta}{d\lambda}, \mu = 0, 1, 2, 3$$

切矢为零的条件如下：

$$g_{\mu\nu} v^\mu v^\nu = 0$$

对于最简单的黑洞——施瓦西黑洞，其周围的时空是球对称的，因此每一束光经过其的轨迹都可以简单地在过黑洞中心的一个平面上计算。考虑用 $r(\lambda), \theta(\lambda) = \frac{\pi}{2}, \phi(\lambda), t(\lambda)$ 计算，则化简后得到：

$$\frac{d^2}{d\phi^2} \left(\frac{1}{r} \right) = -\frac{1}{r} + 3M \left(\frac{1}{r} \right)^2$$

本项目直接采用半隐式欧拉法进行数值求解。

2.2 体渲染

仅考虑吸积盘的发光和散射，不考虑星光，原因是星光的相对强度太低（类似于观测太阳就看不到星星）。那么直接使用体渲染公式：

$$I = \int_0^S P(x) \exp \left[- \int_0^x \alpha(t) dt \right] dx$$

$P(x)$ 是 x 处吸积盘三通道发光功率密度， $\alpha(x)$ 是 x 处对光的吸收率。公式中不考虑多散射和背景光。

2.3 黑体辐射

从 <https://www.freesion.com/article/8583280768/> 中借鉴了温度到 RGB 三通道颜色的算法（实际上更正确的做法可能需要转换为 XYZ 坐标，这是未来的改进方向之一）。目前还考虑了伽马校正，先将颜色各分量作 2.2 次方，叠加后再作 $1/2.2$ 次方。

2.4 吸积盘噪声

2.4.1 Perlin 噪声

Perlin 噪声是在每个整数点 \mathbf{u}_i 上给定一个梯度向量 \mathbf{v}_i , 然后每个 \mathbf{x} 上的取值通过:

1. 通过一个连续的坐标变换 A 变换到 \mathbf{y}
2. 计算:

$$\sum_i f(\mathbf{y}, \mathbf{u}_i) (\mathbf{y} - \mathbf{u}_i) \cdot \mathbf{v}_i$$

u_i 是 \mathbf{y} 附近的整数点, $f(\mathbf{y}, \mathbf{u}_i)$ 是权函数。将结果作为 \mathbf{x} 处的噪声强度 $perlin(\mathbf{x})$ 。

2.4.2 旋转吸积盘

将时间 t 传入 shader, 于是每次计算 \mathbf{x} 处的密度函数就采用:

$$perlin(\mathbf{A}(\mathbf{x}, t))$$

其中 A 是一个变换, 用于寻找 x 位置处的吸积物质在 t 时刻前处于什么位置。这里处理地非常简单, 直接使用 A 为一个仅于 $|x|$ 有关的旋转变换 (不同位置有不同的角速度)。施瓦西时空下物质转速是:

$$v(r) = \sqrt{\frac{GM}{r - r_s}}$$

其中 $r_s = 2M$

2.4.3 双盘叠加

由于不同位置角速度不同, 旋转久了会产生扭曲, 于是我借鉴了 <https://zhuanlan.zhihu.com/p/20536269771> 的方法, 即用两个吸积盘叠加。我采取的方式是:

对于盘 1, 其强度正比于 $\sin^2 \pi(kt - \text{floor}(kt))$, 其用于计算吸积盘的 t 代为: $t - \frac{1}{k}\text{floor}(kt)$

对于盘 2, 其强度正比于 $\cos^2 \pi(kt - \text{floor}(kt))$, 其用于计算吸积盘的 t 代为: $t - \frac{1}{k}\text{floor}(kt - 0.5)$

这样，两个盘在相位突变的时候强度都为 0，叠加总强度为 1，视觉效果较好。

2.5 多普勒效应

对于同一个运动的物体，不同位置观测者观测到的光频率、强度均不同。相对论下运动物体辐射强度、频率和其本征参考系下的强度、频率之间关系：

$$I = \frac{I_0}{\gamma^4(1 + \beta \cos \theta)^4}$$

$$\nu = \frac{\nu_0}{\gamma(1 + \beta \cos \theta)}$$

再结合黑洞积吸盘某处的转速：

$$v(r) = \sqrt{\frac{GM}{r - r_s}}$$

其中 $r_s = 2M$

考虑之前的黑体辐射公式，温度的变化比与 γ 的变化比相同.

2.6 泛光

先提取高亮的部分. 这里为了防止突变，我将低亮部分也取了一部分（采取权值为 $(\frac{p}{p_0})^k$, p_0 是临界值）。之后利用横向高斯卷积和纵向高斯卷积交替进行（如果直接采用二维卷积核，那么每次卷积计算量为 N^2 , 现在这样做每次计算量为 $2N$ ）.

2.7 其他时空下的测地线

平直时空下线元长度表达式：

$$ds^2 = -dt^2 + dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2$$

之前采用的施瓦西时空，其中线元长度表达式为：

$$ds^2 = -(1 - \frac{2M}{r})dt^2 + (1 - \frac{2M}{r})^{-1}dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2$$

这说明其是球对称的，因此其很容易化为一个仅与 $r - \phi$ 相关的微分方程，很容易求解。而旋转不带电黑洞产生的时空 (Kerr 时空) 中的线元长度表达式为：

$$ds^2 = - \left(1 - \frac{2Mr}{\rho^2} \right) dt^2 + \frac{\rho^2}{\Delta} dr^2 + \rho^2 d\theta^2 + \left(r^2 + a^2 + \frac{2Mra^2 \sin^2 \theta}{\rho^2} \right) \sin^2 \theta d\phi^2 - \frac{4Mra \sin^2 \theta}{\rho^2} dt d\phi$$

其中：

$$\rho^2 = r^2 + a^2 \cos^2 \theta, \quad \Delta = r^2 - 2Mr + a^2.$$

接下来我们面临两个困难：

1. 线元形式峥嵘而崔嵬，而联络有 64 个，每个的计算式：

$$\Gamma_{\mu\nu}^\lambda = \frac{1}{2} g^{\lambda\rho} (\partial_\mu g_{\nu\rho} + \partial_\nu g_{\mu\rho} - \partial_\rho g_{\mu\nu})$$

虽然联络具有对称性 $\Gamma_{\nu\gamma}^\mu = \Gamma_{\gamma\nu}^\mu$ ，我们也要至少计算 32 次上面的式子。无论如何，对如此复杂的度规求两位数次数的偏导、矩阵乘，在计算上是很难的。

2. 在球坐标下，我们在 $\sin \theta = 0$ 时会遇到奇点，如果数值方法较差我们会面临困难。而且，球坐标的不均匀性可能需要我们采用自适应的步长... 总之由球坐标带来的困难很多。

我采用的解决方法如下：

问题 1：通过 python 的 sympy 库进行符号计算，然后用其 ccode 功能把符号计算结果导出到 .h 的 cpp 头文件。

问题 2：解决方案是在笛卡尔坐标系 $Oxyz$ 下进行计算。但是我们没有直接计算由 xyz 显示确定的联络，而是计算了由 r, θ, ϕ, t 给定的非平直时空球坐标联络（直接符号计算及其费时），再减去平直时空在 r, θ, ϕ, t 下的联络得到 $\tilde{\Gamma}$ 。可以证明， x, y, z, t 对 λ 的二阶导，可以直接通过由假联络 $\tilde{\Gamma}$ 在 r, θ, ϕ, t 下计算的坐标的二阶导坐标变换回来（把计算得到的 $(r'', \theta'', \phi'', t'')$ 当作矢量）。（如果直接用球坐标下的联络计算，我们还需要面临坐标变换带来的加速度。这里取假联络实际上相当于用平直时空联络全 0 的特征把坐标变换带来的加速度消掉了。）

3 实现流程

3.1 测地线测试 GEODESIC_TEST

本项目主要用于检测头文件”geodesic.h”(用于计算施瓦西时空下的测地线)。

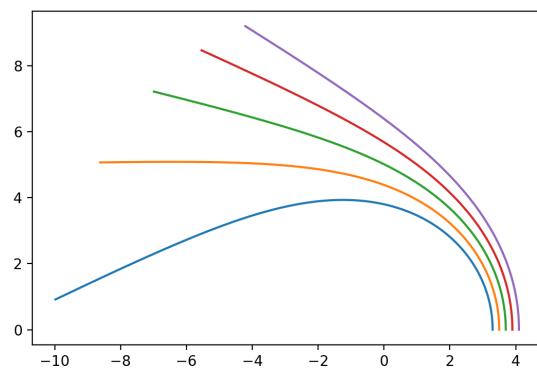


图 1: 类光测地线测试结果, 其中黑洞半径 $R=2$

3.2 体渲染测试 VOLUME_R

在 cpu 中实现了体渲染流程 (volume_render.cpp), 用 GSL 计算了氢原子的轨道密度函数并显示 (使用平直时空下的步进 Flat_ST_RayMarch)。

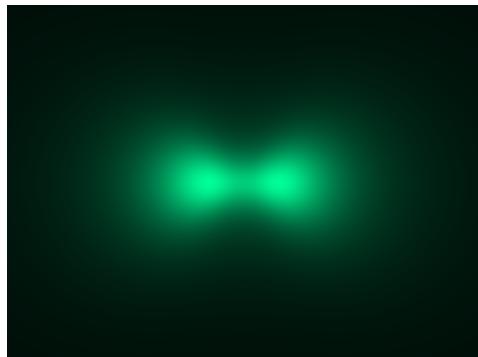


图 2: 主/角/磁量子数 $3/2/2$

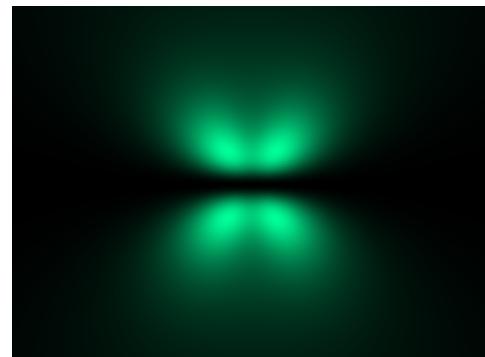


图 3: 主/角/磁量子数 $3/2/1$

之后用 Schwarzschild_BH_RayMarch 类 (用到了 geodesic.h) 初步计算

了施瓦西黑洞的效果。

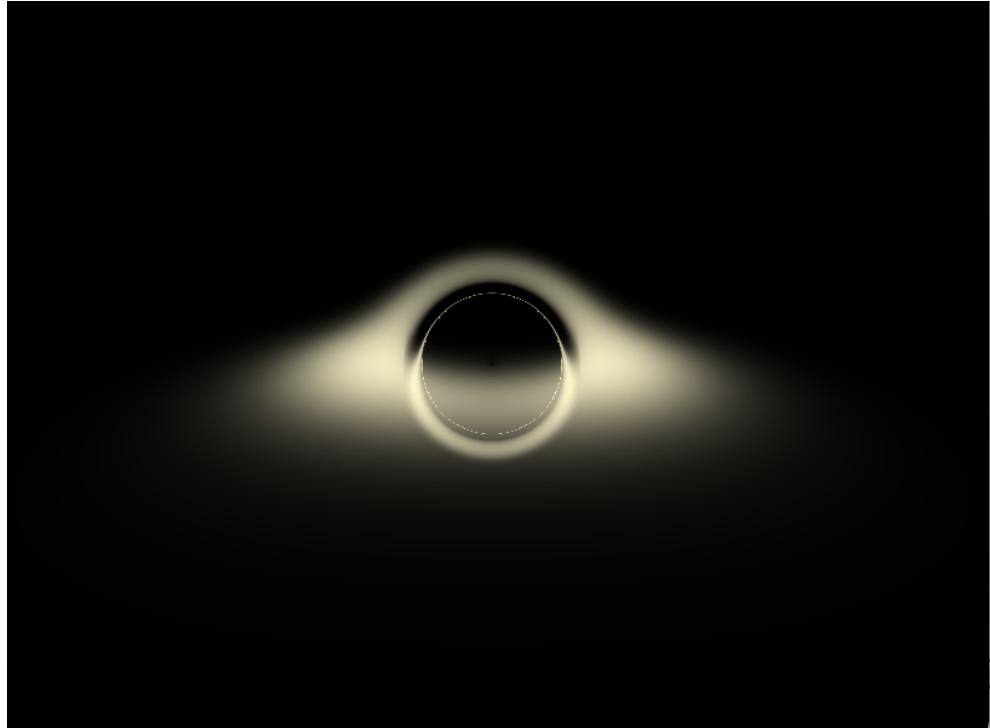


图 4: 带褐色吸积盘的施瓦西黑洞

此时成图颇具廉价感。从现在的角度来说：缺少泛光、吸积盘没有噪声、颜色不正确。

3.3 CPU 上的泛光

为了方便测试、兼容之前的工作，我决定现在 CPU 上实现泛光。一开始直接使用二维高斯核进行卷积，但效率感人。后来用横纵一维卷积交叠代替了高斯卷积，效率逐渐可以接受。（直到此时，所有工作成果都不是实时的，要花很长时间渲染一张图）。再略调了调颜色使吸积盘变成橙色后，得到了如下结果：

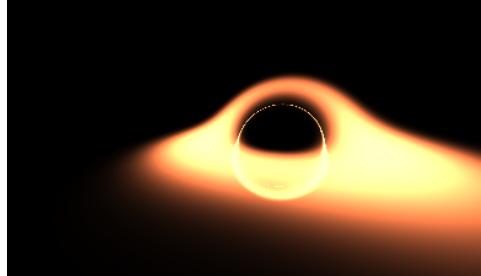


图 5: 效果图 1

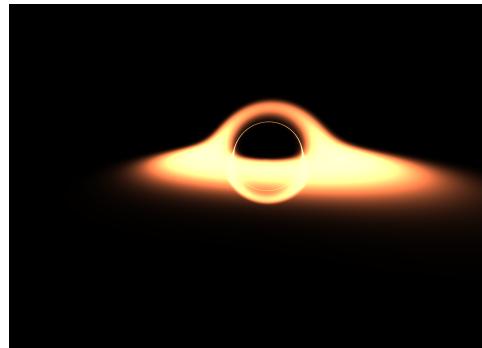


图 6: 效果图 2

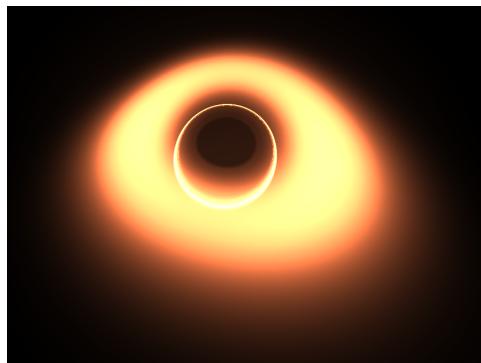


图 7: 效果图 3

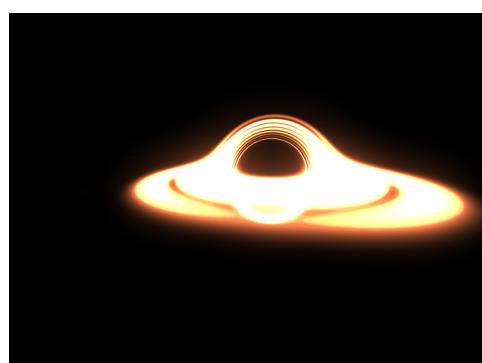


图 8: 效果图 4(在吸积盘函数中加了和 r 相关的正弦函数, 另外泛光有点过高了)

3.4 初步并行、吸积盘噪声、光强多普勒效应

这里抛弃了 Volume_Render.h, 开始使用 opengl 的 compute shader 进行体渲染。添加了吸积盘噪声 (使用 Perlin 噪声), 并且添加了光强多普勒效应。

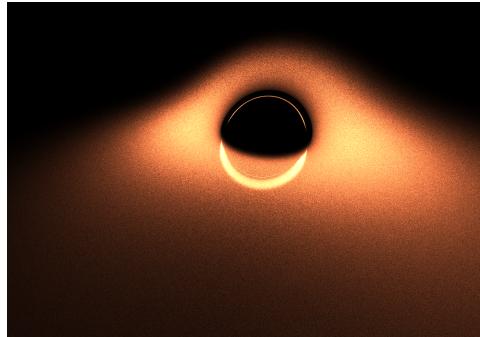


图 9: 效果图 1(无多普勒, 细噪声)

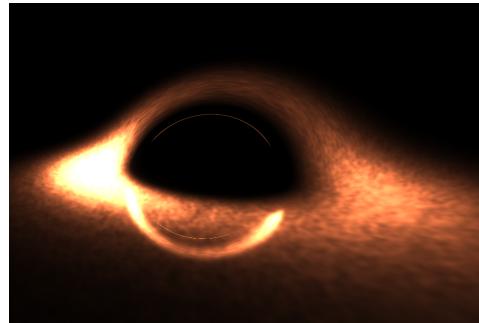


图 10: 效果图 2(光强多普勒, 粗噪声)

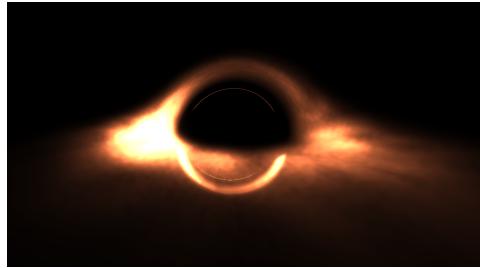


图 11: 效果图 3(光强多普勒, 粗噪声)

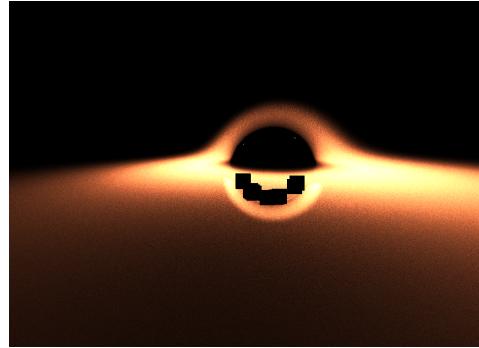


图 12: 遇到的小 bug(似乎是雾生成时对负数进行了开跟)

3.5 温度-RGB 转换、频率多普勒、泛光 GPU 化

在之前的 compute shader 中添加了温度到 RGB 的转换以及频率的多普勒效应。之后将泛光的计算也移到了 GPU 上。



图 13: 效果图 1(无多普勒)

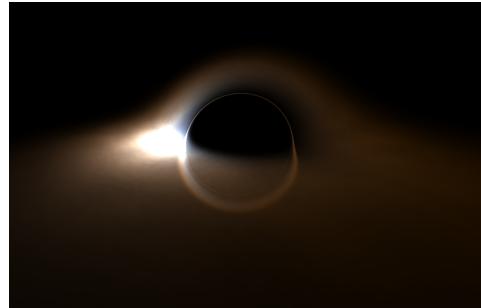


图 14: 效果图 2(双多普勒)



图 15: 效果图 3(高温, 双多普勒)



图 16: 效果图 4(高温, 双多普勒)

3.6 参数可调、旋转吸积盘

之前的工作都是在固定参数下进行的, 想改效果只能从源码修改, 且渲染很长时间 (主要花在光线步进) 后只能得到一张图。现在我将 ImGui 引入程序, 实现可调参数。并且通过将时间传入 shader 的方法实现让每帧吸积盘呈现旋转样式, 让图变成动画。

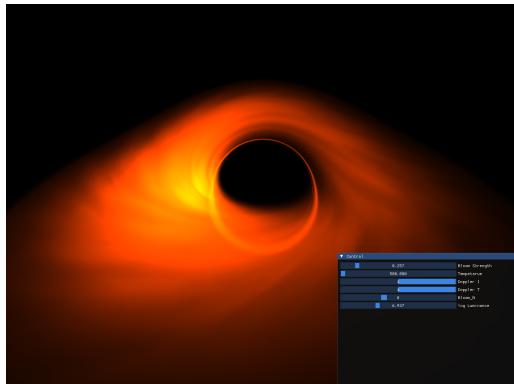


图 17: 效果图 1

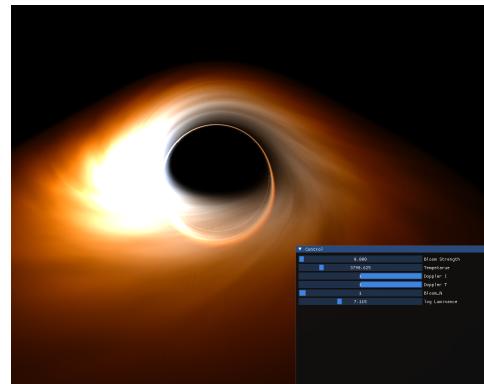


图 18: 效果图 2

3.7 Kerr 黑洞、其他时空

《星际穿越》电影中的黑洞是一个旋转的黑洞 (Kerr 黑洞)，因此我决定实现 Kerr 黑洞的渲染。使用 sympy 符号化计算导出了 Kerr 时空的 64 个联络，之后用笛卡尔坐标系光线步进。得到一系列图：

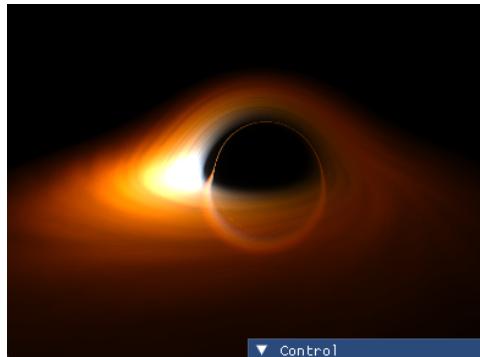


图 19: 温度参量 2500K, 黑洞自旋
 $a=0.9$

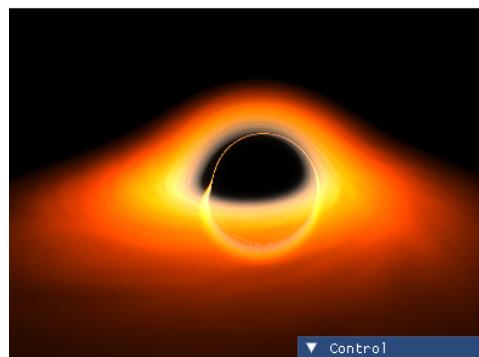


图 20: 温度参量 2500K, 黑洞自旋
 $a=0.9$ (无多普勒)

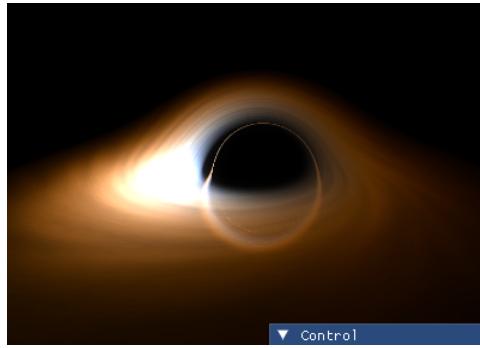


图 21: 温度参量 6000K, 黑洞自旋
 $a=0.9$

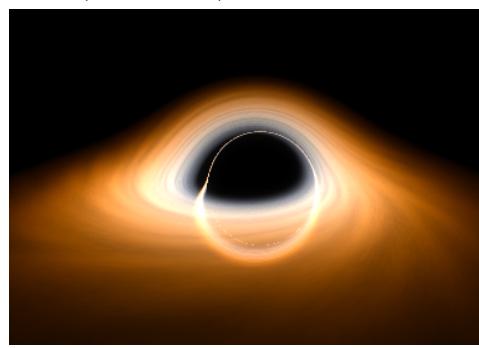


图 22: 温度参量 6000K, 黑洞自旋
 $a=0.9$ (无多普勒)

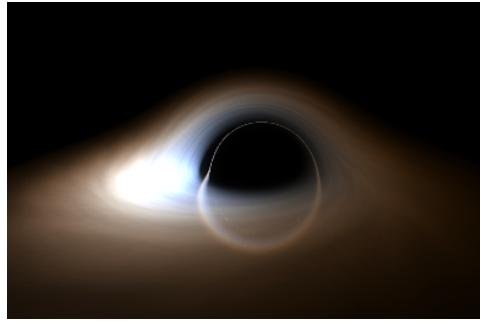


图 23: 温度参量 10000K, 黑洞自旋
 $a=0.9$

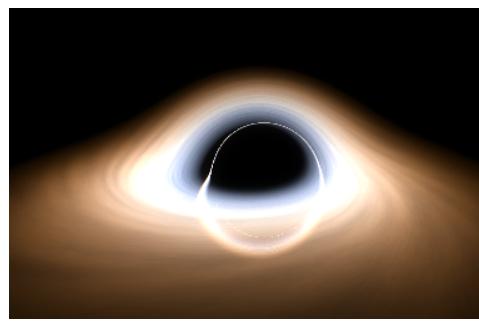


图 24: 温度参量 10000K, 黑洞自旋
 $a=0.9$ (无多普勒)



图 25: 温度参量 14000K, 黑洞自旋
 $a=0.9$

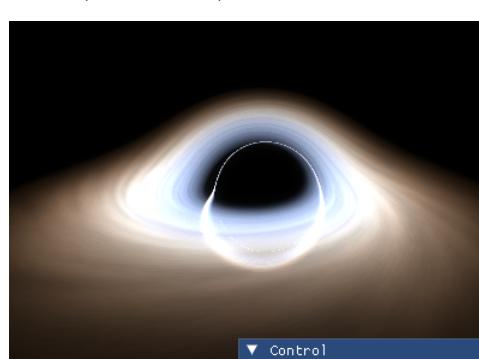


图 26: 温度参量 14000K, 黑洞自旋
 $a=0.9$ (无多普勒)

以下是我的黑洞渲染器的结果，于《星际穿越》中的卡冈图雅黑洞进行对比。差异主要在泛光强度、吸积盘噪声设置、吸积盘温度设置。

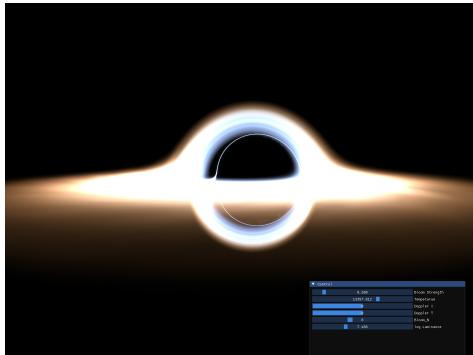


图 27: 本渲染器:13500K,a=0.9, 无多普勒

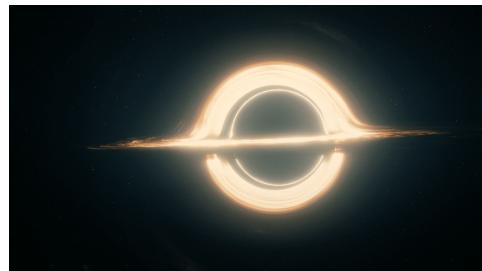


图 28: 星际穿越: 温度自旋未知, 无多普勒

为提升视觉效果的下一步工作：更改吸积盘噪声、增加外圈物质吸收率。

原则来说带电黑洞的渲染也可以实现，但由于其形式和施瓦西黑洞类似，视觉效果也应当相近，这里没有实现。

此外，我还测试了两个目前在物理上被认为不可能出现于宇宙中的时空：

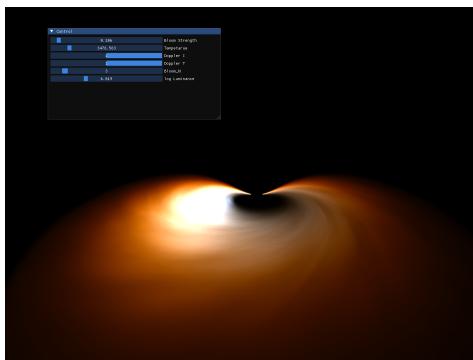


图 29: 负值量施瓦西“黑洞”



图 30: 自旋大于 1 的 Kerr 黑洞, 自旋 a=1.5

1. 其是负质量，这意味着其会对光线进行排斥，这也是为什么其顶上会出现一个楔形黑色区域：发射向那里的光都被排斥到吸积盘外。

2. 其是自旋 >1 的黑洞，原则上会出现裸露的奇点。不过由于光线步进速度较慢以及设置保护性的截断（防止对负数开跟、分母由正向负突变），目前其中心的细节需要进一步改进。

4 下一步工作

4.1 GPU 并行光线步进

目前由于封装和简单考虑，光线步进程序仍然是串行/CPU 并行（提升不高，约 30%）的。下一步工作是将光线步进程序并行化，尝试进行实时渲染（可变摄像机位置）。

4.2 吸积盘噪声更改

目前的噪声细节度还不够，下一步工作是将噪声细节度提升到更高的水平，达成更好的视觉效果。

4.3 辐射-RGB 转换

目前的 RGB 转换是直接基于经验公式给出的，下一步工作是将其改为基于辐射强度的转换（即将辐射强度转换为 XYZ 坐标系下的颜色，再转换为 RGB）。

4.4 其他时空

目前的工作仅限于施瓦西黑洞和 Kerr 黑洞，下一步工作是实现其他时空下的测地线计算（如 Reissner-Nordström 黑洞、Kerr-Newman 黑洞等）。并观测更多非物理的时空。

4.5 说明文档完善、可移植性、轻量化

目前的说明文档较为简陋，下一步工作是完善说明文档、使其更易于阅读。另外，当前的代码并不易于移植（需要用户用 CMake 配置、编译），

而且有部分参数必须在程序中调试。那么之后要使其更易于移植 (如使用 CMake 配置文件)。甚至可以考虑将其打包为一个可执行文件。此外，应该把测试中的 GSL 去掉，其存储占用了 40MB 左右的空间，且 GSL 的使用并不多。GLM 更是没有用到，也可以考虑删除。