



计算机图形学

任博

渲染框架: <https://gitee.com/nankaigraphics/nrenderer>

课程评分

一揽子评分计划：

- 课堂表现 <10%（课上问答，作业，出勤等）【填充至 90 分以上部分开根号后向上取整】
- 论文阅读报告 <20%（提交总结 PPT）【最低占比 10%】
 - 前沿科研成果的论文阅读报告 OR 光栅图形学知识的介绍性报告，不含章节标题 15 张 PPT 以上
 - 必须是提炼总结，非复述照抄。
- 课程作业 >70%（提交报告及代码）【最高占比 90%】
 - 大作业 A1,A2 任选，小作业一个

课程作业

□ 共通要求：

- 作业报告（Report）：各 10% - 清晰的书面组织，完善的样例演示，可以**直接**编译运行的源代码

□ A1: 光线跟踪核心算法改进（大作业 1，截止日期第 17 周）

- 基本要求（BASE 40%）：（必须基于所提供的演示程序，）对光线跟踪的核心算法进行改进和开发。可选不限于：增加新的核心渲染算法；提供更好的光源种类、模型支持；编写加速结构显著改进渲染速度；优化采样算法……视改进幅度可能获得升级分（BONUS）不超过 20%。**除数学工具函数外不允许使用任何库提供的现成实现。**

- 给分示例：**新增**核心渲染算法至少 40%

- 实现了 Metropolis 渲染方法——得 60%（然而不建议，截止目前尝试成功率 0%，挑战者可选）

一篇讲 Metropolis 的文章：https://www.pbr-book.org/3ed-2018/Light_Transport_III_Bidirectional_Methods/Metropolis_Light_Transport#

- 仅增加了对 environment map 类型光源的支持——得 5%

- 实现了光子映射渲染方法——得 40%+

- 通过 kd-tree 等手段的优化使原算法性能提升了 2-10 倍——得 10%-40%

- 提供了优化的采样方法使相同质量效果的总渲染时间缩短到 50%-10%——得 10%-40%

课程作业

- ▮ A2：实现一个二维网格流体模拟器（大作业 2，截止日期第 17 周）
 - ▮ BASE：50% - 1）基于课程内容实现二维 Staggered Grid 网格上的流体求解器 2）能够提供模拟结果（速度场或密度场变化）的直观演示 3）除数学工具函数外不得使用已有的库提供的代码
 - ▮ BONUS：每项升级获得额外 10%，如：场景中包含运动的固体；提供三维场景求解器等

课程作业

- B: 使用计图（jittor）框架实现 Conditional GAN（小作业，截止日期**第 14 周**）
- BASE：10% - 1）实现好网络；2）使用至少一个通用图像库训练并测试效果，训练集与测试集不能包含相同图像；3）对结果进行一定的数据统计，并在报告中进行分析。

课程作业

- 每小组包含 1 至 2 人，从 A 中任选一（或二）项，与 B 合并计算总分。
- 大作业分数计算公式：

$$Score = \sum (Report + Base + Bonus)$$

阅读报告部分建议内容

- ▢ 光栅图形学中的复数个话题（PPT 中介绍复数个话题）
 - ▢ 直线与圆弧的扫描转化
 - ▢ 多边形的扫描转化与填充
 - ▢ 裁减
 - ▢ 消隐
 - ▢ 反走样
- ▢ ACM SIGGRAPH, EuroGraphics, Pacific Graphics 最近十年内的文章
- ▢ 曲线曲面造型与参数化技术中的话题
- ▢ 注意：
 - ▢ 在阅读报告开始注明所阅读的文献。正文每页 PPT 须言之有物，按整体充实程度给分。
 - ▢ 重点报告内容：文献所做的事情；它使用了何种方法，为什么（一定）要使用这种方法；为达到目的文献设计了怎样的程序流程：其中各步骤如何处理了什么事情，原因又是什么。
 - ▢ 截止日期第 14 周

关于大作业

图形学涉及知识面太广

- 本来认为图形学大作业只需要了解自己需要做的内容就好，例如我的目标就是光子映射的话，我只需要了解光子映射的内容
- 然而光子映射是基于光线追踪框架的，要了解光子映射，我就得去学习光追的内容
- 而光追的第一步是光线投射，所以还得学习光线投射
- 而光线在三维空间中所有的投射也好，反射折射都和物理有关，所以需要了解相关的物理学知识
- 而物理又是用数学来描述的，所以我又不得不复习丢了很多年的立体几何以及向量计算
- 要写kdtree做加速，我还顺带写了两个数据结构作业，1. 最大堆 2.kd tree
- 网上博客良莠不齐，还得回去看作者的论文以及其他一手资料，又是一场英文学习
- 代码框架是c++11标准的，让我们这些98用户刚开始也很头疼，一个auto让一切都困难了许多，这甚至是c++课程
-

以上都还只是冰川一角，这个作业堪称我本科生涯最硬核的课之一，让我净工作时间至少达到了104小时。其中阅读了两篇论文，一篇photon map作者写的实现指南（76页）的大部头著作以及无数篇鱼龙混杂的网上博客.....

啃源码很重要(RTFSC)

- 首先第一条是一个小小的建议：
 - 作业框架下一届使用的时候请助教辛苦一下添加注释🙏
 - 作业框架下一届使用的时候请助教辛苦一下添加注释🙏
 - 作业框架下一届使用的时候请助教辛苦一下添加注释🙏

- 这个源码，上手的时候根据那有限的提示和珍贵的几行注释，完全不明白要从哪里开始我们浩大的工程
- 但是同时这次经历也告诉我：就算没有一行注释，先尝试理解并自己添加注释，然后马上着手开始实现自己的功能。遇到不懂的地方再去啃源码。这样不断递进下去，有目的地阅读源码，就算一行注释也没有，也是最终能理解并“视如己出”的。
- 源码可以帮助我们理解理论：例如许多物理的公式，我本人看的时候很纳闷全是积分微分这可怎么编程，可以说完全不理解。但是公式是为了更好的描述一个现象，编码的时候可能会将其拆解，变成更好理解的方式。因此阅读代码，也能够学习到如何去理解一个理论。
- 源码很重要。

拒绝畏难情绪

其实很多身边选择图形学课的同学都放弃了代码框架，因为没有注释大家都有畏难情绪，再加上期末周，更是热锅上的蚂蚁。

说实话第一天看框架我也有心理防线被攻破的感觉，感觉想放弃。

其实代码读到后面越来越顺心，甚至帮我实现了一些很难实现的内容。例如我可以直接调用框架中光线求交的函数，这个自己实现估计也是需要个半天来理解 + debug的。

因此，学计算机的人，需要顶住压力，因为，连编译原理和计算机组成原理都熬过去了，图形学不过是最后一个“程序员三大浪漫”的最后一个而已，感谢这学期的图形学课程。我感觉我计算机人对艰难险阻的探索已经几近完善，越过了这三座大山，以后任何任务都不再畏惧。或许以后的科研路很艰难，但是这番修行，肯定锻造了我相对平和的心态。

再次感谢图形学这门课程，也感谢老师带我们入门。

有关最新技术发展的说明

- 允许使用 ChatGPT, GPT4 等手段对撰写代码或撰写报告提供支持
 - 若使用上述 AI 手段，必须在报告中详细叙述所涉及的内容（如涉及何种函数或模块等）、目的、效果，以及详细操作过程。
 - 满足以上条件，则不影响分数（即正常按作业实现内容、效果评分），否则一旦被发现使用 AI，按抄袭处理。

如果发现抄袭、作假

- ❑ 抄袭、作假的形式包括但不限于：
 - ❑ 抄袭他人小组成果
 - ❑ 从网上的博客文章，代码站等地复制代码
 - ❑ 复制他人的文章
- ❑ 我们都会很遗憾

程序架构照着写
核心函数网上抄
百度结果第一页
代码相同还俩人
学习终是为自己
多技总比蹉跎香
劝君稍微想一想
搜索基础谁更强