

# 西安交通大学计算机图形学实验文档

## 大作业

作者：罗思源 李昊东 张博闻

组织：计算机图形学课题组

时间：November 28, 2022




# 目录

<b>第 1 章 实验要求</b>	<b>1</b>
1.1 概述	1
1.2 实验报告	1
<b>第 2 章 渲染方向：全局光照</b>	<b>2</b>
2.1 实验简介	2
2.2 实验内容	2
2.2.1 Generating Camera Rays	2
2.2.2 Intersecting Objects	3
2.2.3 Path Tracing	4
2.2.4 Materials	5
2.3 现场验收	5
2.3.1 验收前的准备	5
2.3.2 验收的整个流程	6
2.4 报告提交	7
2.4.1 报告中应包括的内容	7
2.4.2 报告的打分标准	7
<b>第 3 章 几何与建模方向：曲面简化</b>	<b>8</b>
3.1 实验简介	8
3.2 实验内容	8
3.2.1 基于 QEM 的曲面简化	8
3.2.2 参考资料	8
3.3 现场验收	9
3.4 报告提交	9
<b>第 4 章 物理模拟方向：更真实准确的小球运动模拟</b>	<b>11</b>
4.1 实验简介	11
4.2 实验内容	11
4.2.1 通过外力实现可变加速度	11
4.2.2 实现四种不同的时间积分	11
4.2.3 碰撞中的能量损失与摩擦力	12
4.3 现场验收	12
4.3.1 验收前的准备	12
4.3.2 验收的整个流程	12
4.3.3 验收的打分标准	13
4.4 报告提交	13
4.4.1 报告中应包括的内容	13
4.4.2 报告的打分标准	13

# 第 1 章 实验要求

## 1.1 概述


 **笔记** 我们希望大家在完成大作业的过程中体验查找资料（尤其是英文资料）、理解算法、转化实现的过程。在此过程中，我们将尽力提供资料渠道、编程技巧、代码调试等方面的帮助，但自选实验的文档将不会像之前一样面面俱到。

自选实验可以从渲染、几何、物理三个方向中任选其一，每个方向有各自的具体评分标准，我们将尽量保证得分难度相近。完成实验后要求进行验收并提交一份实验报告。完成实验的过程中，大家可以：

- 参考教材、讲义、课程、博客、视频等解释性的资料
- 有限度地参考开源代码，但必须如实说明所参考代码的来源，并且**严禁直接复制代码**

实验按 20 分制打分，其中验收和实验报告各占 10 分。

## 1.2 实验报告

 **笔记** 我们希望能合理控制大家的负担，反对过度增加实验报告长度的行为。提交一份事无巨细的冗长报告并不会增加你的得分。

请将实验报告定稿导出为 PDF 格式，并与测试数据和所有源代码打包为 ZIP 格式一并提交。正确的提交目录结构如下：

```
[班级-学号-姓名]
├── [班级-学号-姓名].pdf
├── src (即 Scotty3D/src 这个目录)
│   ├── ... (若干源代码文件)
├── data
│   ├── ... (若干测试数据)
```

将此目录压缩为 [班级-学号-姓名].zip 后，提交至思源学堂。班级专业（如“计算机”或“计试”）和班号（如“001”）间不应该有空格，各部分间用英文短横线隔开。

实验报告的统一要求为：

- 包含所选实验、班级、姓名。
- 正文字号为五号（10.5 磅）或 11 磅。
- 文末列出自己所用的全部参考资料。报告中所写的观点、理论、算法和使用的图片，如非自己完成则必须列于参考资料中。不要求特定的格式，但要求同类别参考资料（网页、著作、论文等等）的条目格式相同。

一份包含这些信息的实验报告最终得分不会低于 4 分：

- 个人信息
- 算法简述（即使只有文字）
- 如实地记录了自己的实验结果（如果并没有成功完成实验，那么记录自己的进展和中间结果，即使是错误的结果）

在成功完成实验的基础上，即使临近期末时间紧张，写一份 6 分以上的实验报告还是相当容易的，请大家不要放弃记录自己的实验。

每个方向的具体要求参见后续章节说明。

## 第2章 渲染方向：全局光照

### 2.1 实验简介

本实验旨在通过模拟光线在复杂场景中的弹射过程得到具有真实感的图像，希望通过本次实验让同学们对全局光照算法有一个更深入的了解。同学们需要用到的基础知识包括线性代数，概率论以及 C++ 编程，在实现路径追踪的过程中我们不需要从头开始实现，我们将在 Scotty3D 上完成路径追踪算法。

### 2.2 实验内容

本实验共包括 4 个子实验，分别是 Scotty3D A3:Pathtracer 中的 Generating Camera Rays, Intersecting Objects, Path Tracing 和 Materials，同学们需要依次完成这四个实验，其中 Materials 是选做实验。下面对以上四个实验做简单介绍，详细内容见 Scotty3D/docs/pathtracer/内的说明。

#### 2.2.1 Generating Camera Rays

本实验的任务是从相机位置发出射线，为后续的路径追踪算法做准备。本实验中，在相机坐标系下相机在原点，相机朝向 z 轴负方向，成像平面距离原点 1 个单位长度（成像平面为  $z=-1$ ），具体见图 2.1。

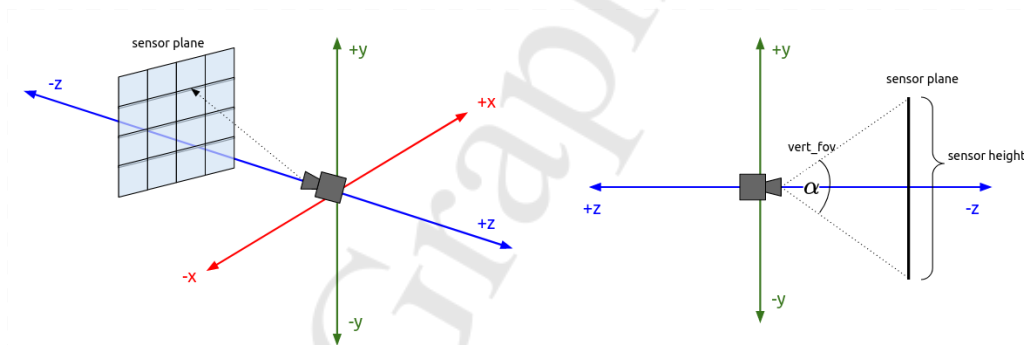


图 2.1: 相机坐标系

本实验需要补全的函数包括：

1. Camera::generate\_ray (在 src/student/camera.cpp 中)，该函数的输入是 2 维屏幕坐标  $(x,y)$ ，其中  $x,y \in [0,1]$ ，返回结果为世界坐标系下的射线。一种推荐的实现思路是先在相机坐标系下计算射线的起点和方向，然后将射线转换到世界坐标系下。
2. Rect::sample (在 src/student/samplers.cpp 中)，在长方形内均匀随机采样 2 维点坐标。
3. Pathtracer::trace\_pixel (在 src/student/pathtracer.cpp 中)，在该函数内实现对像素的超采样。

在完成该实验后，我们需要验证结果的正确性。我们需要在 Pathtracer::trace\_pixel 中添加如下代码：

```
if(RNG::coin_flip(0.0002f)){
    log_ray(ray, 10.0f, Spectrum(0.1, 0.2, 0.3));
}
```

添加这部分代码后，我们要对整个工程进行编译，打开 Scotty3D，导入一个场景，开始渲染（先点击 Open Render Window，在弹出的页面中点击 Start Render），渲染结束后点击 Draw rays，观察由相机发出的射线位置是否正确。

以 cbox 场景为例，产生的射线如图2.2所示：

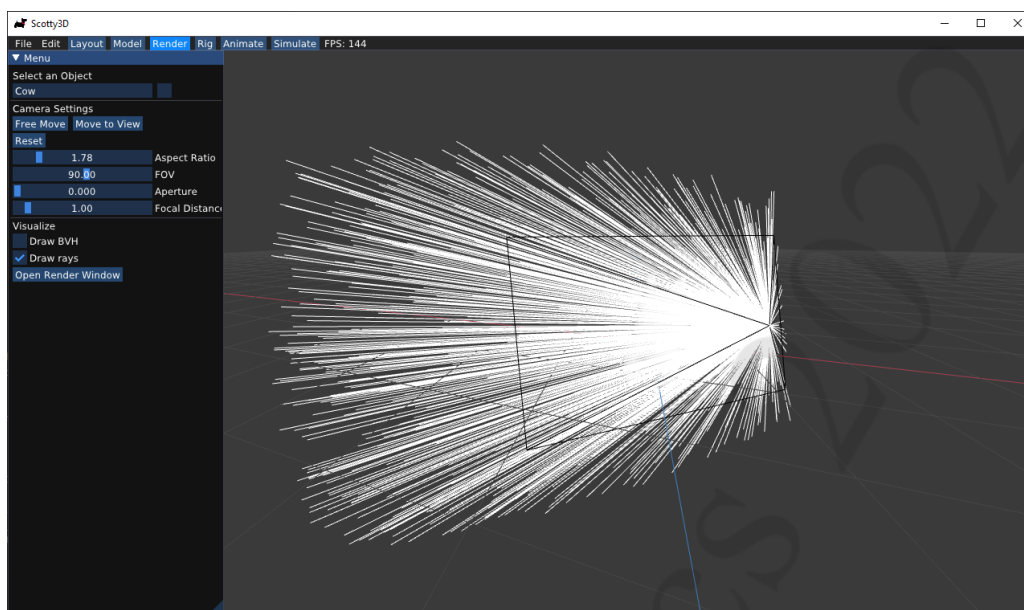


图 2.2: 相机射线参考结果

## 2.2.2 Intersecting Objects

本实验的任务是求射线与场景中物体交点。在 Scotty3D 中物体的类型包括三角形和球形，同学们需要分别实现射线与三角形和射线与球形的求交点算法。

本实验需要补全的函数包括：

1. `Sphere::hit` (在 `src/student/shapes.cpp` 中)，该函数在球形的局部坐标系下求解球形与射线的交点，球心位于原点。该函数输入为球形局部坐标系下的射线，返回结果是相交的结果。推荐同学们参考[ray-sphere-intersection](#)或者B站视频 [GAMES101 \(P13 31min10s-36min45s\)](#) 来实现本算法。需要注意的是，每条射线都有一个距离区间  $[dist\_bounds.x, dist\_bounds.y]$ ，交点与射线起点  $O$  的距离应该在距离区间内。当一条射线与球形有两个交点的时候，而第一个交点与射线起点  $O$  的距离不在距离区间时，应该选择第二个交点。如图 2.3所示。

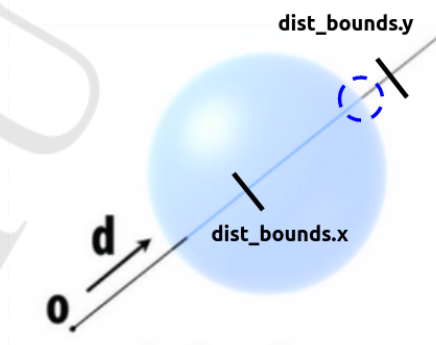


图 2.3: 球形与射线求交点的特殊情况

2. `Triangle::hit` (在 `src/student/tri_mesh.cpp` 中)，该函数在世界坐标系下求解三角形与射线的交点。该函数输入为世界坐标系下的射线，返回结果是相交的结果。推荐同学们参考[Moller-Trumbore algorithm](#)或者B站视频 [GAMES101 \(P13 38min40s-52min45s\)](#) 来实现本算法。需要注意的是，如果射线与三角形相交，交点处



的法线方向应该由交点处的重心坐标求得：

$$\vec{n}_v = \alpha \vec{n}_a + \beta \vec{n}_b + (1 - \alpha - \beta) \vec{n}_c \quad (2.1)$$

其中  $\vec{n}_v$  为交点处的法向， $\vec{n}_a$ ， $\vec{n}_b$ ， $\vec{n}_c$  分别为三角形顶点的法向， $\alpha$  和  $\beta$  为三角形的重心坐标。

完成该部分实验后，导入场景（media/cbox\_lambertian.dae），得到的法向渲染结果应该如图 2.4 所示：

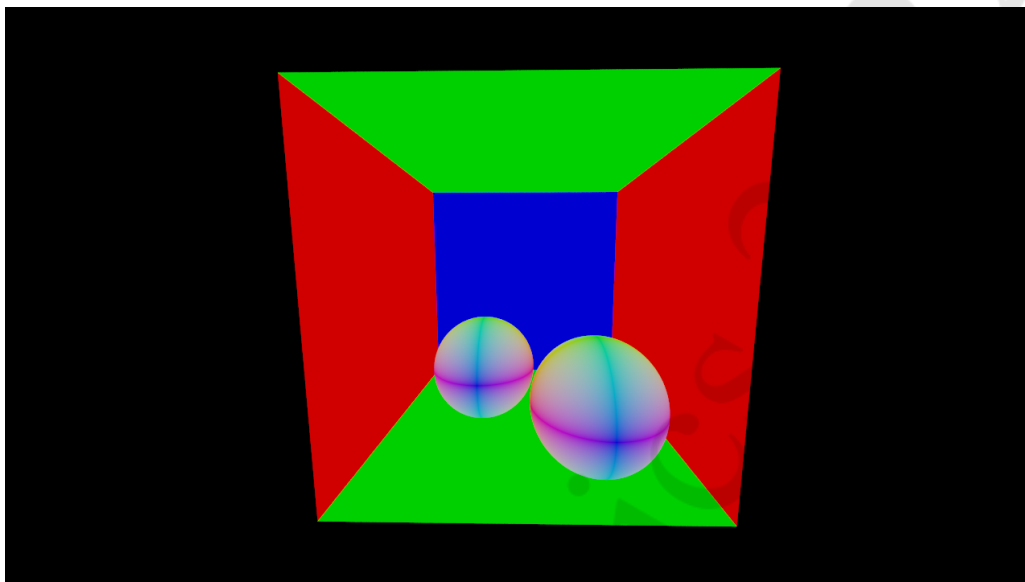


图 2.4: cbox\_lambertian 的法向渲染参考结果

### 2.2.3 Path Tracing

在完成 Generating Camera Rays 和 Intersecting Objects 后，目前的渲染器仅可以计算物体的可见性。本实验的任务是模拟光线最终进入相机前在场景中复杂的弹射路径。模拟光线的多次弹射对于渲染图片的真实感十分重要，需要注意的一点是在 src/student/pathtracer.cpp 中的所有函数是在世界坐标系下进行计算的，在 src/student/bsdf.cpp 中的所有函数是在射线与物体交点处的局部坐标系下进行计算的，交点为局部坐标系下的原点，交点处面积微元的法向为  $\vec{n} = (0, 1, 0)$ 。

本实验需要补全的函数包括：

1. Pathtracer::trace（在 src/student/pathtracer.cpp 中），我们需要仔细阅读该函数并对其进行修改，去掉返回物体表面法向颜色的代码。
2. BSDF\_Lambertian::scatter（在 src/student/bsdf.cpp 中），根据光线的出射方向，对光线的入射方向进行采样。
3. BSDF\_Lambertian::evaluate（在 src/student/bsdf.cpp 中），返回入射光与出射光能量的比值。
4. BSDF\_Lambertian::pdf（在 src/student/bsdf.cpp 中），计算采样得到的出射光线的概率密度。需要注意的是，本实验中我们采用的是 cosine-weighted sampling，具体信息如图 2.5 所示。
5. Pathtracer::sample\_indirect\_lighting（在 src/student/pathtracer.cpp 中），对非直接光照来说，光线最终到达着色点前在场景中至少弹射一次。
6. Pathtracer::sample\_direct\_lighting（在 src/student/pathtracer.cpp 中），对于直接光照，光线在到达着色点之前不经过任何弹射。

完成该实验后，导入场景（media/cbox\_lambertian.dae），在每个 pixel 上采样 1024 条射线（Samples=1024），每条射线的最大深度为 8（Max Ray Depth=8），然后开始渲染，正确的渲染结果应该如图 2.6 所示。

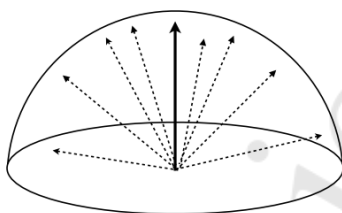
## Example—Cosine-Weighted Sampling

**Cosine-weighted hemisphere sampling in irradiance estimate:**

$$f(\omega) = L_i(\omega) \cos \theta \quad p(\omega) = \frac{\cos \theta}{\pi}$$

$$\int_{\Omega} f(\omega) d\omega \approx \frac{1}{N} \sum_i^N \frac{f(\omega)}{p(\omega)} = \frac{1}{N} \sum_i^N \frac{L_i(\omega) \cos \theta}{\cos \theta / \pi} = \frac{\pi}{N} \sum_i^N L_i(\omega)$$

**Idea: bias samples toward directions where  $\cos \theta$  is large (if  $L$  is constant, then these are the directions that contribute most)**



CMU 15-462/662

图 2.5: Cosine-Weighted Sampling

### 2.2.4 Materials

本实验的任务是实现镜面反射与折射效果。

本实验需要补全的函数包括：

1. Pathtracer::sample\_indirect\_lighting 和 Pathtracer::sample\_direct\_lighting (均在 src/student/pathtracer.cpp 中), 与 Path Tracing 中的漫反射不同, 折射和反射不使用蒙特卡洛积分。这时我们需要 BSDF::is\_discrete 来判断我们是否需要进行蒙特卡洛积分。
2. reflect 和 BSDF\_Mirror::scatter (在 src/student/bsdf.cpp 中), 实现光的反射。
3. refract 和 BSDF\_Glass::scatter (在 src/student/bsdf.cpp 中), 实现光的折射。需要注意的是, 当光线由高折射率的物体射向低折射率物体, 射线的入射角度过大时, 会发生全反射现象。折射时光线的能量损失见 **PBR-book: 8.2.3-Specular Transmission** 中的公式 (8.7)。

完成该部分实验后, 导入场景 (media/cbox.dae), 在每个 pixel 上采样 1024 条射线 (Samples=1024), 每条射线的最大深度为 8 (Max Ray Depth=8), 然后开始渲染, 正确的渲染结果应该如图 2.7 所示。

建议在实验的过程中, 按照图 2.8 的过程逐步进行实现。

## 2.3 现场验收

在完成该实验后, 助课程教将进行现场验收, 现场验收时只验收实验 1-3, 不验收实验 4。

### 2.3.1 验收前的准备

1. 至少完成实验 1, 实验 2 和实验 3。
2. 对自己实现的算法进行复习, 确保在被问到某一函数的具体实现过程时可以对照代码快速作答。

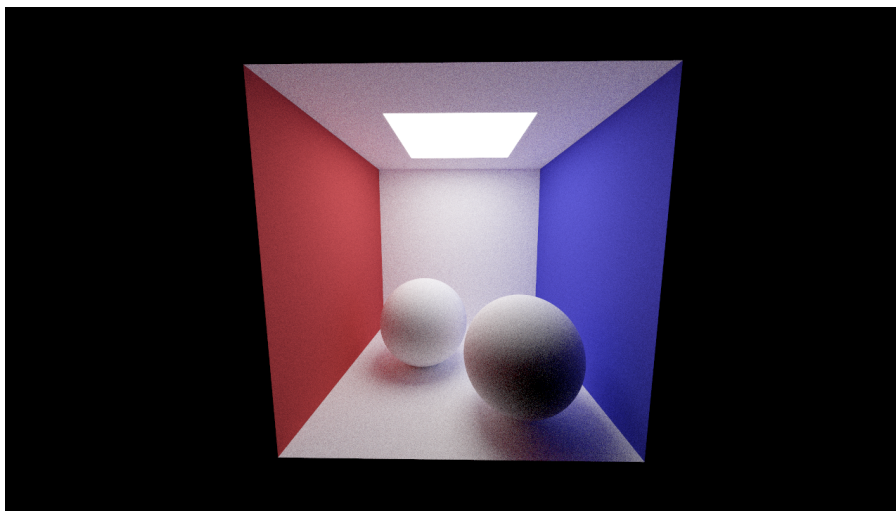


图 2.6: 漫反射参考结果

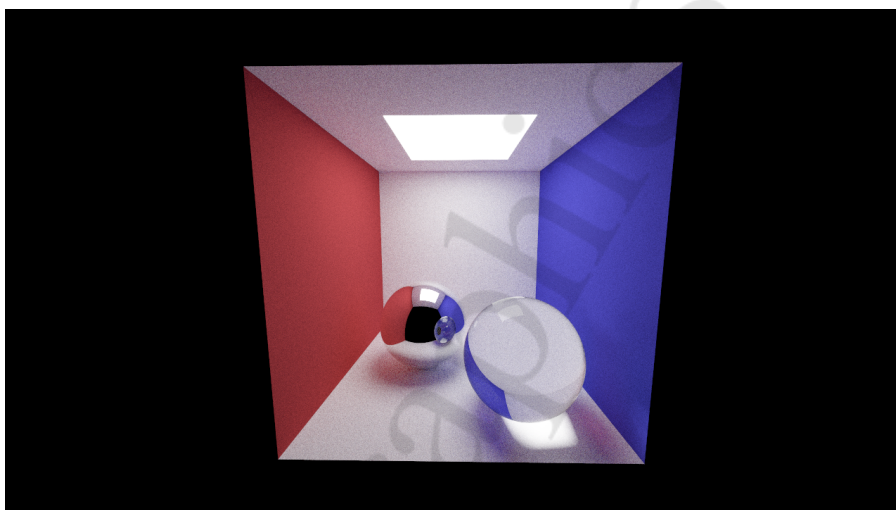


图 2.7: 折射与反射参考结果

3. 可以准备纸质版或电子版的笔记，在验收时可以查看笔记，根据笔记的提示对回答助教的问题。

### 2.3.2 验收的整个流程

验收时，每位同学有 5 分钟时间，现场渲染实验 3 的参考结果，在等待渲染的过程中助教将对实验 1-3 中涉及到的知识点和代码细节进行提问，根据同学们的现场作答进行评分，助教会问三个问题，答对两个问题即可。验收满分为 10 分，实验结果分占 6 分，回答问题占 4 分。

如下是助教可能进行提问的问题，同学们可以提前进行准备：

1. 对照着 `Camera::generate_ray`，讲一下实现的逻辑。
2. 为什么要在每个像素上发射多条射线？
3. 对照着 `Triangle::hit`，讲一下实现逻辑。
4. 对照着 `Sphere::hit`，讲一下实现逻辑。
5. 描述蒙特卡洛积分如何应用于 cosine-weighted sampling 的半球采样。
6. 路径追踪算法的整体流程。



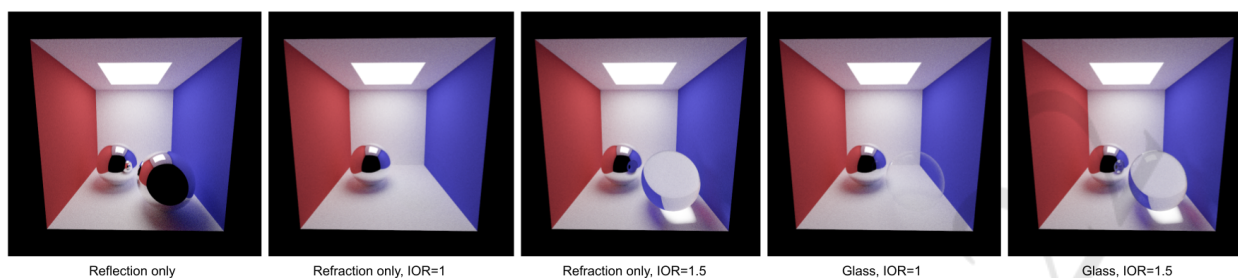


图 2.8: Cbox Debug

## 2.4 报告提交

除了现场验收，大家要提交相应的实验报告。实验 4 为选做实验，如果有同学选做实验 4 需要在报告中包括有关实验 4 的内容。

### 2.4.1 报告中应包括的内容

报告至少应该包括如下内容：

1. 对路径追踪算法进行描述。
2. 对补全的函数的逻辑进行简单描述。
3. 简述用到的算法（射线与三角形、球形求交点，蒙特卡洛积分，cosine-weighted sampling 等）。
4. 实验得到的结果。
5. 列出所有的参考材料。

### 2.4.2 报告的打分标准

未选做实验 4 的同学报告基础分为 10 分，选做实验 4 的同学报告基础分为 11 分，我们将检查报告中的全部内容并按照如表 2.1 所示的方法在基础分上进行加减。最终报告得 10 分即为满分。

表 2.1: 打分标准

情况	加减分值
没有对路径追踪算法的描述或者描述错误	-1
没有对所补全函数的逻辑进行描述	-1
没有简述用到的算法	-1
报告中没有包括实验结果或实验结果错误	-1
实验报告中包括实验过程中遇到的困难以及解决方案	+0.25
没有列出实验所用到的参考资料	-1
实验报告中包括这次实验的心得体会以及对实验课程的建议	+0.5
实验报告排版工整	+0.25

## 第3章 几何与建模方向：曲面简化

### 3.1 实验简介

所谓曲面简化，是指给定一个面数较多的网格（同样是三角流形网格），在尽可能保持原有形状的前提下减少顶点、边和面。本实验基本采用与 lab2 一致的思路，借助半边网格这一数据结构，用局部操作和全局操作实现基于 QEM 的曲面简化算法。

### 3.2 实验内容

#### 3.2.1 基于 QEM 的曲面简化

简化网格的算法不止一种，这里采用的是基于二次误差度量 (QEM, Quadric Error Metrics) 和边折叠操作的简化算法。这个算法用**折叠边**的方法进行简化，每次将一条边“折叠”起来，两端的顶点合并为一个，原先连接到两端点的边也都连接到这个顶点。用边折叠操作简化网格时有两个关键点：

- 决定折叠哪一条边
- 合并两个端点产生新顶点的位置

二次误差度量可以定量估算顶点到原始网格（曲面）的距离。开始简化前估算出每条边最佳的折叠位置（新顶点）和这一点到原网格的距离（折叠代价），简化过程中用优先队列动态维护所有边的代价，并每次选择代价最低的边进行折叠，直至面数或顶点数达到要求为止。

你需要填补 `src/student/meshedit.cpp` 中的 `Halfedge_Mesh::simplify` 函数和所需的局部操作 `Halfedge_Mesh::collapse_edge`。边折叠操作的具体说明参见我们提供的 Scotty3D PDF 文档，这里额外强调两个注意事项：

- 折叠操作是**不安全的**，折叠一个流形网格的某一条边之后，它可能不再是一个流形网格！为此，我们必须在执行折叠之前判断这次折叠是否会破坏流形性质。设两个端点分别是  $v_1, v_2$ ，它们各自的邻接顶点集合分别是  $\mathcal{N}(v_1), \mathcal{N}(v_2)$ ，仅当

$$|\mathcal{N}(v_1) \cap \mathcal{N}(v_2)| = 2$$

时，这次折叠才不会破坏流形性质。

- 折叠三角形面的某一条边将导致这个三角形面退化成两条重叠边，此时应当从网格中删除这个面（调用 `erase` 方法）。由于我们的题目是简化三角形网格，你无需考虑四边形或多边形面的情况。

#### 3.2.2 参考资料

任何关于 C++ 语法的问题请首先参考 [en.cppreference.com](http://en.cppreference.com)（或中文翻译版本 [zh.cppreference.com](http://zh.cppreference.com)）。由于 Scotty3D 使用 C++ 17 语法，从其他平台上查找资料时请注意时效性，谨慎对待 2017 年之前的资料。

关于曲面简化算法，推荐的参考资料包括：

- Scotty3D Developer Manual: Simplification，详见我们提供的 PDF 文档。
- Surface Simplification Using Quadric Error Metrics  
[www.cs.cmu.edu/~./garland/quadrics/quadrics.html](http://www.cs.cmu.edu/~./garland/quadrics/quadrics.html)
- GAMES 102 第 19 课：曲面简化（视频 P11）<https://www.bilibili.com/video/BV1NA411E7Yr>

### 3.3 现场验收

请大家在验收前准备好自己的材料，包括：

- 以下三个问题的答案：
  - 完成实验的过程中参考了哪些资料？请复述它们的来源、形式，但无需像论文的参考文献那样详细。
  - 简述自己实现的算法。我们鼓励大家自由地选择重点或省略细节，禁止像背书一样直接背诵网络博客的解释。
  - 实现算法的过程中在哪些步骤进行过较多的调试？出错的原因是什么？如果你未曾遇到任何明显的阻碍，请解释你的算法实现中最关键的局部流程或数据结构。
- 已经加入 Simplification 算法的 Scotty3D 程序，我们将现场观看你的运行结果，实验结果在视觉上应当是合理的。

演示自己的程序时请现场清空所有编译的中间文件和可执行文件，重新编译并运行，最终的结果**没有明显的视觉问题**即认定为通过，演示结果和回答问题共占 10 分。

验收时每位同学有**五分钟**的时间回答问题，允许携带任何形式的讲稿和提示但不允许全程读稿，超时即停。这些问题都没有标准答案，我们也不要求大家的回答如何精妙，能够真实反映自己完成实验的过程、没有严重的逻辑错误即不会被扣分。

### 3.4 报告提交

实验报告基准分为 10 分，最终得分为基准分加（减）评审分值。几何与建模方向的实验报告应当依次包含如下内容：

1. 对算法流程的简述。请在**半页至一页 A4 纸**的范围内简述算法，必须包含核心的公式或图解，禁止成段地复制或翻译参考的资料。
2. 对程序效率的分析或探究。请从这三个问题中选择一个，将解释写入报告：
  - 请给出推导计算复杂度的简要过程。
  - 请尝试思考此曲面简化算法并行化的可能性，并简要描述并行部分的数据依赖关系、同步与互斥关系。
  - 用能够记录时间的库（例如 `ctime`）和较大规模的数据（建议数万至数十万个面）测试并记录程序各部分的执行时间，寻找占用时间最多的部分，并尝试从计算复杂度、内存连续性等方面解释原因。
3. 自己测试数据的来源和网格的大致规模。若测试数据为自制，请说明自己生成测试数据所用的工具并给出过程证明；若为网络下载，请注明下载来源。
4. 测试自己程序的测试结果。大多数算法以测试用例的截图展示即可，若算法的结果中有关键性的定量细节，请以表格形式写在截图之后。
5. 所有参考资料。

我们将检查报告中不满足上述要求的部分，并按照如下标准评审：

表 3.1: 打分标准

问题	加减分值
算法流程简述部分少于半页或多于一页	-2
算法流程简述部分只有文字描述	-1
缺少分析程序效率的部分	-1
在思考并行化可能性的基础上，能够进一步给出算法并行化的实现方案（说明技术细节，但无需代码实现）	+1
在测试分析程序效率瓶颈的基础上，提出改进 Scotty3D 中的半边网格或简化算法的思路（可以是数据结构、存储方式等），增强内存连续性或降低计算复杂度。	+1
自制数据没有给出生成工具和过程证明，或下载数据没有给出下载来源	-1
测试用例少于 3 个	-1
同类别参考资料的格式不一致	-1

## 第 4 章 物理模拟方向：更真实准确的小球运动模拟

### 4.1 实验简介

本实验旨在 lab3 的基础上，在小球运动与小球碰撞的处理上进行逼近物理世界的计算，以达到更真实的效果，希望能通过该实验让同学们对物理模拟中的时间积分，离散化以及更真实的碰撞响应有更深入的了解。同学们需要用到的基础知识包括线性代数，微积分以及 C++ 面向对象编程。本实验需要基于 lab3 的基础完成的，来实现更精细的物理模拟。

### 4.2 实验内容

本实验共包括 3 个小实验，分别是 1. 通过外力来实现可变的加速度；2. 通过四种不同的时间积分来实现更准确的运动；3. 在碰撞的响应中引入能量损失和摩擦力。同学需要完成这三个实验，并在 Scotty3D 中的左侧 Menu 中加入相关的可视化交互工具来实现类似如下参数的调节。（比如：外力大小、外力的作用时间或作用范围、小球的质量、时间积分的选取类型和碰撞摩擦力相关系数等）。可视化交互示例具体见图 4.1。

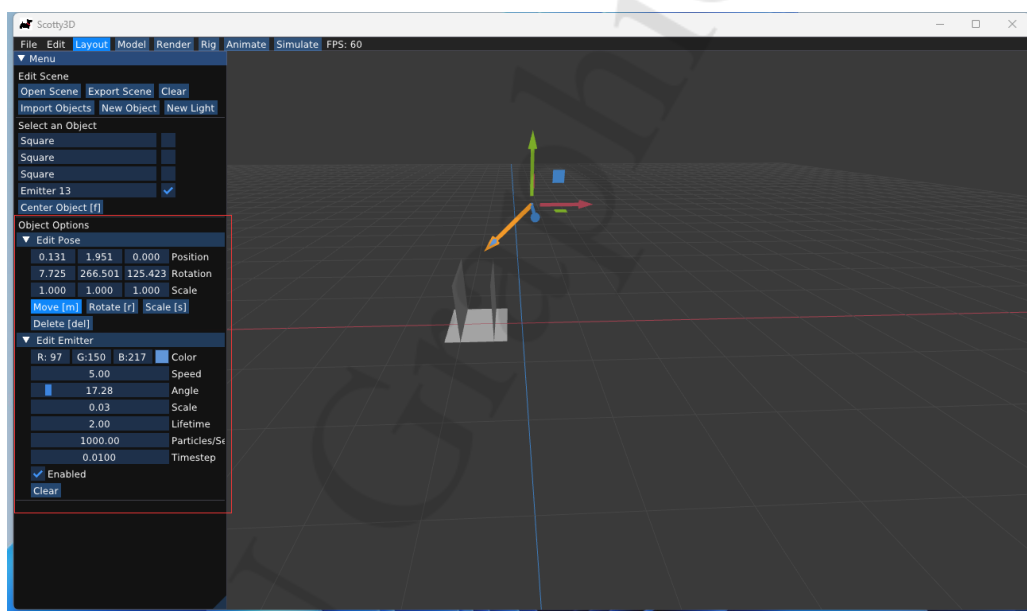


图 4.1: 参考示例

#### 4.2.1 通过外力实现可变加速度

本实验的任务是在 lab3 中仅有重力加速度的基础上，实现可调节的外力作用。比如在指定区域或指定小球运动时间段中，施加小球一个不同方向于重力的外力，从而实现类似于“风场”的效果。实质上是对小球加速度大小和方向的修改，可以在 lab3 修改的 `update()` 函数中完成。完成后需要实现在 GUI 界面中可调的可视化接口。

#### 4.2.2 实现四种不同的时间积分

在时间离散化后，时间积分的实现对小球的运动起着重要的作用。一个优秀的时间积分可以使大步长的计算中系统仍然保持稳定。在 lab3 中，由于加速度不可变，所以最简单的时间积分实现——“显式时间积分”



便可以在 `timestep` 不大的时候有一个较好的效果。而在受外力影响加速度可变的情况下，更稳定的时间积分则是需要被实现的。该实验需要同学们实现四种不同的时间积分，如下。

1. Forward Euler Integration(已经实现)
2. 4-th Runge-Kutta Integration(可以参考[该链接](#)去实现)
3. Implicit (Backward) Euler Integration
4. Symplectic Euler Integration(可以参考[该链接](#)去实现)

同学们也可以参考[这个资料](#)，更深入地从数学层面理解不同时间积分的物理意义。

在 lab3 中已经 `update()` 函数中实现了第一种时间积分，后三种也可以在 `update()` 中实现。而同学们需要在实现四种积分之后，在 GUI 界面中使用开关的形式来指定某一种时间积分的调用与否，以此来控制小球的运动。

### 4.2.3 碰撞中的能量损失与摩擦力

在 lab3 的实验中，当检测到碰撞后，下一步进行的操作是修改碰撞位置的速度，修改后的速度方向沿法线方向对称，速度大小不变，即完全弹性碰撞。但真实世界中的碰撞，由于能量损失，摩擦力等等因素，并非是完全弹性的。那么实验就需要通过一些新的模型去计算摩擦力和能量损失，并将这种现象更新到实验里碰撞后速度的修改中。这里的具体实现方法可以大家尽情调研实现，不做硬性算法层面的要求。

但需要满足：可以通过范围是  $[0,1]$  的一个或几个系数来对能量损失或者摩擦力大小进行定义和调节。随着系数的变换，小球在碰撞之后也会体现出不同的运动趋势。而这些系数需要具有一定的数学或物理含义，并能够解释得通。在实现后，系数也需要在 GUI 界面中实现可视化的可调接口。

## 4.3 现场验收

在完成该实验后，助课程教将进行现场验收，现场验收时验收实验 1-3。

### 4.3.1 验收前的准备

1. 完成实验 1，实验 2 和实验 3。
2. 对自己实现的算法进行复习，验收时需要检查 `update()` 的实现，问到相关算法或者 GUI 接口实现的时候，可以对照代码快速作答。
3. 可以准备纸质版或电子版的笔记，在验收时可以查看笔记，根据笔记的提示对回答助教的问题。

### 4.3.2 验收的整个流程

验收时，每位同学有 5 分钟时间，现场运行 Scotty3d 并导入 `particles.dae` 文件或自己定义的模型文件，分别通过实现的可视化交互方式来介绍实验 1-3 的实现和结果，助教将在验收时根据同学们的现场介绍进行评分，同时会问三个问题，答对两个问题即可。

如下是助教可能进行提问的问题，同学们可以提前进行准备：

1. 你是怎么调用 `nanogui` 的接口实现可视化的？
2. 实验 3 中碰撞之后的速度修正是怎么做的？你定义的参数的数学意义是什么？
3. 实验 2 中四种时间积分的优劣是什么？你在实验中感觉哪一种的效果最好？
4. 简要描述一下实验 1 中力的实现思路 and 把力放进 `update()` 的实现方法。

**注意：**如果在展示实验的时候，同学们已经在介绍中清楚说明了上述问题的答案，助教可根据介绍的情况减少问题的数量。

### 4.3.3 验收的打分标准

验收满分为 10 分，得分包括两部分，一部分是实验结果分（6 分），另一部分是回答问题得分（4 分）。对第一部分，三个实验各占 2 分。对于第二部分，回答对一道问题得 2 分，助教会问三个问题。

## 4.4 报告提交

除了现场验收，大家要提交相应的实验报告。介绍自己三个实验具体的工作。

### 4.4.1 报告中应包括的内容

报告至少应该包括如下内容：（对每一个实验分别介绍算法和工程上的实现）

1. 实验环境和简单的最终界面展示。
2. 对补全的函数的逻辑进行简单描述。
3. 简述用到的算法（外力的引入，四种时间积分的离散化形式，计算摩擦力和能量损失的公式以及系数的含义）。
4. 工程上可视化交互界面的实现。
5. 实验得到的结果。
6. 列出所有的参考材料。

### 4.4.2 报告的打分标准

报告的基础分为 10 分，我们将检查报告中的全部内容并按照如表 4.1 所示的方法在基础分上进行加减。最终报告得 10 分即为满分。

表 4.1: 打分标准

情况	加减值
没有描述清楚四种时间积分的原理和实现	-2
没有对所补全函数的逻辑进行描述	-0.5
没有简述用到的算法	-1
报告中没有包括实验结果	-1
实验报告中包括实验过程中遇到的困难以及解决方案	+0.5
没有列出实验所用到的参考资料	-1
实验报告中包括这次实验的心得体会以及对实验课程的建议	+1
实验报告排版工整	+0.5