

# 大作业选题参考

## GAMES101, 2021 年春季

教授：闫令琪

计算机图形学与混合现实研讨会

GAMES: Graphics And Mixed Environment Seminar

发布日期为北京时间 2021 年 9 月 4 日 (星期六) 上午**10: 00**

选题截止至北京时间 2021 年 9 月 9 日 (星期四) 上午**10: 00**

提交截止至北京时间 2021 年 9 月 25 日 (星期六) 上午**10: 00**

---

### 注意：

- 任何更新或更正都将发布在论坛上，因此请偶尔检查一下。
  - 论坛链接：<http://games-cn.org/forums/forum/graphics-intro/>。
  - 你必须独立完成自己的作业。
  - 你可以在论坛上发布帖子求助，但是发布问题之前，请仔细阅读本文档。
  - 在截止时间之前将你的作业提交到 SmartChair 上。
-

## 1 大作业描述

本次大作业的完成时间共计三周，即从**北京时间 2021 年 9 月 4 日 (星期六) 上午 10:00**至**北京时间 2021 年 9 月 25 日 (星期六) 上午 10:00**。在本文档的后续部分，我们给出了一些大作业的参考选题，当然这些仅仅只是参考选题，**我们非常鼓励你根据自己感兴趣的方向自主选题！**要注意的是，如果你是自主选题的话，请在**北京时间 2021 年 9 月 9 日 (星期四) 上午 10:00 前**在 smartchair 平台上提交一份简短的关于你所选题目的描述，我们需要确认你所选题目与图形学的相关程度和难易程度，从而同意你的自主选题。因此，选题描述越早提交越好！选择参考选题的同学，也请在选题截止日期之前上传文件说明自己所选的题目，以便我们进行统计。

对于我们提供的参考选题，我们同时进行了难度等级的估计。请注意，不同的难度等级之间差别并非特别明显（只有难度等级为 3.0 的项目会非常有挑战性）。另外，我们会参考不同的难度等级，期待你有不同的完成度。

大作业要求提交代码、视频与文档。

## 2 选题参考

### 2.1 作业拓展

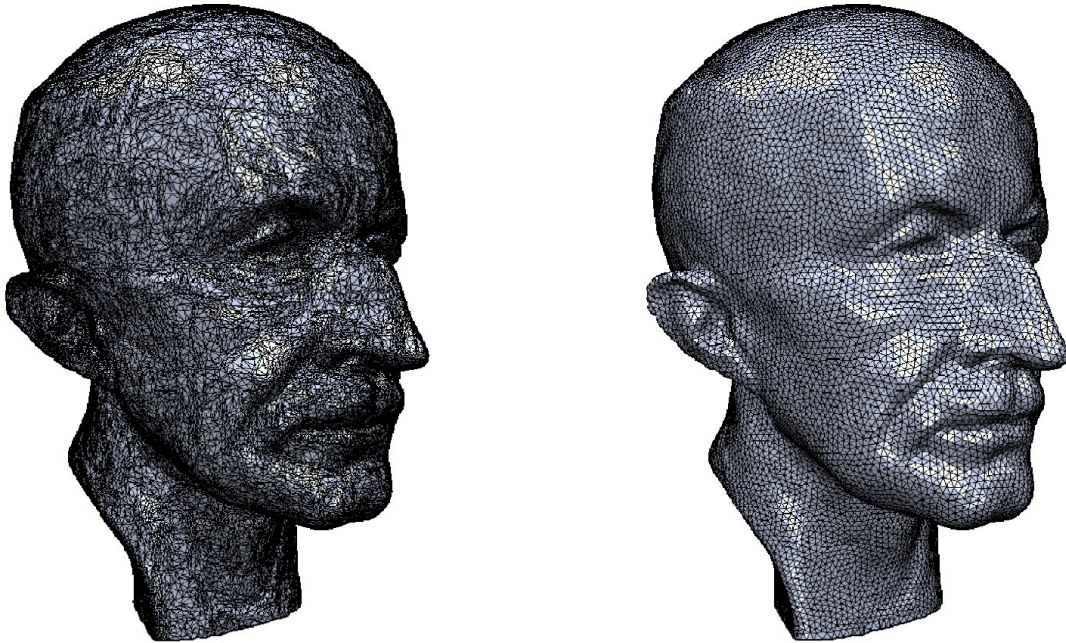
#### 2.1.1 使用 Loop 细分的 Displacement 贴图 (难度等级: 2.0)



在作业中，我们实现了 displacement 贴图以改变对象的几何形状并添加令人惊叹的表面细节。但是，为了捕获更加精细的细节，我们需要具有非常小且多的三角形网格。为了解决这个问题，我们可以使用动态的细分方法。根据 displacement 贴图指定的信息，仅在需要时才将大的多边形细分。之后，将动态细分的网格发送到图形管道中。在此项目中，你可以在 CPU 中把需要细分的三角形使用 Loop 细分，然后使用任何渲染方法将它显示出来，而不需要使用 GPU 编程。

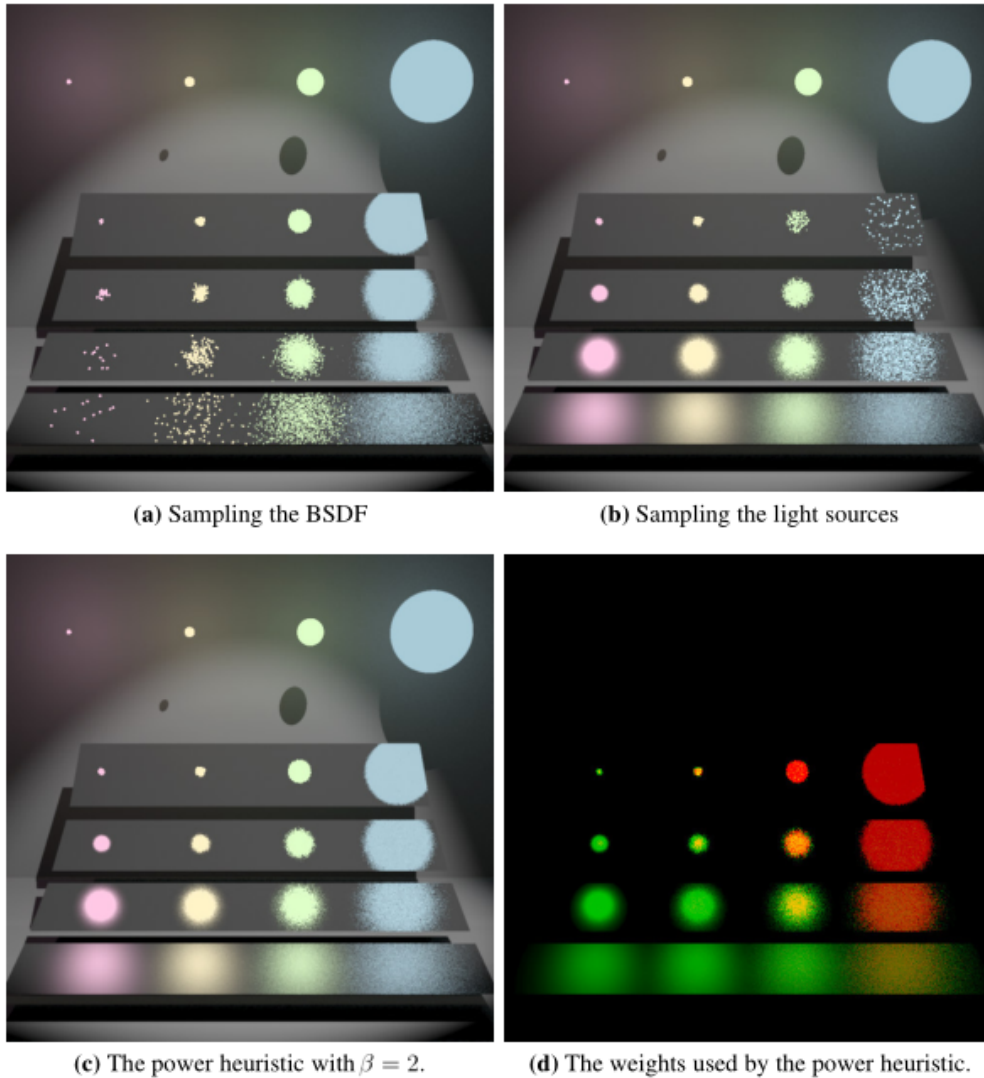
[参考资料](#)

### 2.1.2 网格简化及重新网格化 (难度等级: 1.5)



有时，三角形网格中的三角形数量远远超过所需要的，从而浪费了存储空间，并且会花费很多的计算时间。为了解决这个问题，我们可以简化网格，在其中找到可以使用较少三角形的区域，然后相应地简化这些区域。另外，还可以找到同一表面更好的离散表示。将网格转换为更好的表示形式称为重新网格化。注意这里由于单一方法实现难度较低，我们要求你实现网格简化与重新网格化两种方法。两种方法不要求同时应用。

### 2.1.3 多重重要性采样 (难度等级: 2.0)



课程中提到我们可以对光源采样，也可以对 BSDF 采样，那么两种方法怎样才能结合起来？另外结合两种方法又有什么优势？你可以探索多重重要性采样的方法，当渲染表面的粗糙度较低时，更多依赖于对 BSDF 的采样，而当光源较小时，更多依赖于对光源的采样。注意这里要求你先了解并实现对 BSDF 的重要性采样，再考虑多重重要性采样。

[参考资料](#)

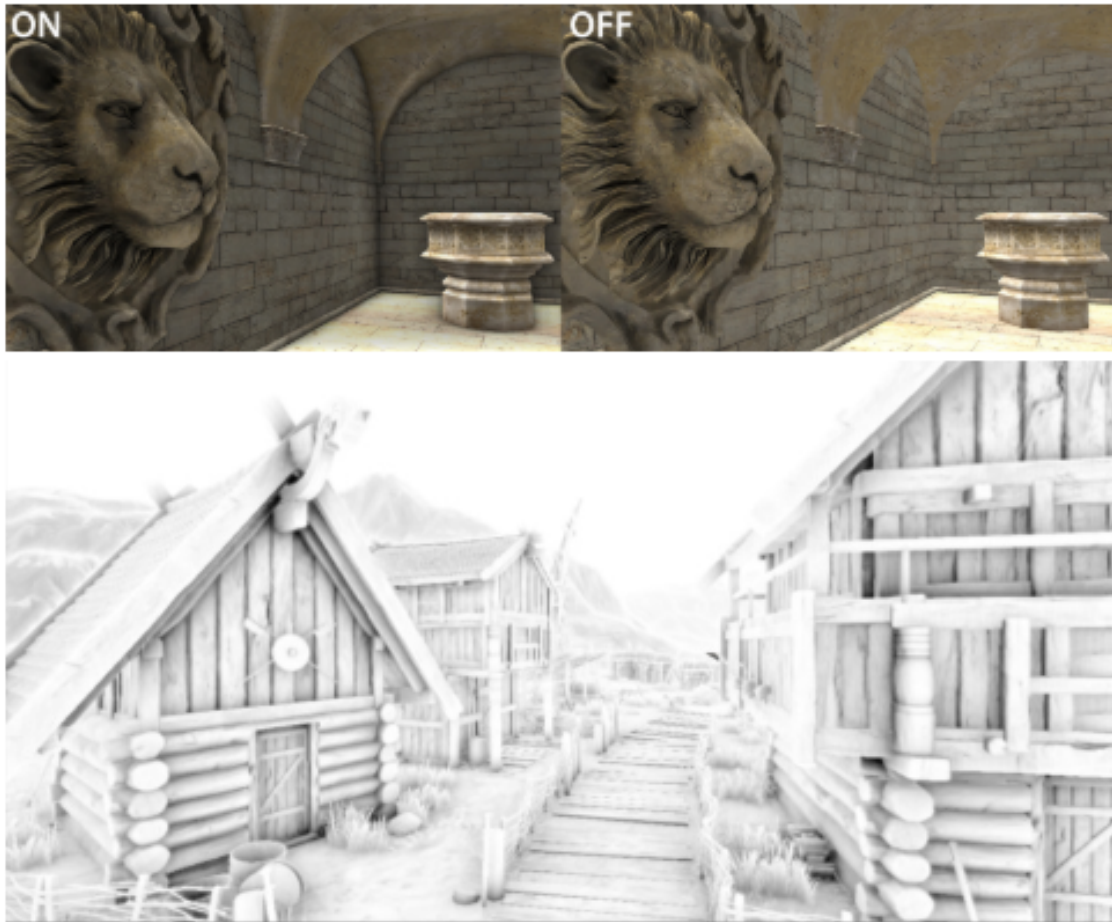
#### 2.1.4 支持光场相机 (难度等级: 1.5)



光线追踪场景时，我们使用针孔照相机或薄透镜模型。我们可以扩展它以模拟光场相机的效果——原来的每个像素现在变成一个网格，记录来自不同方向的radiance。使用此网格，即使场景已经渲染完成，我们也可以对场景进行实时的重新聚焦，而不需要重新发射光线。你可以渲染一些光场图像，并实现一个漂亮的GUI来展示你的结果！



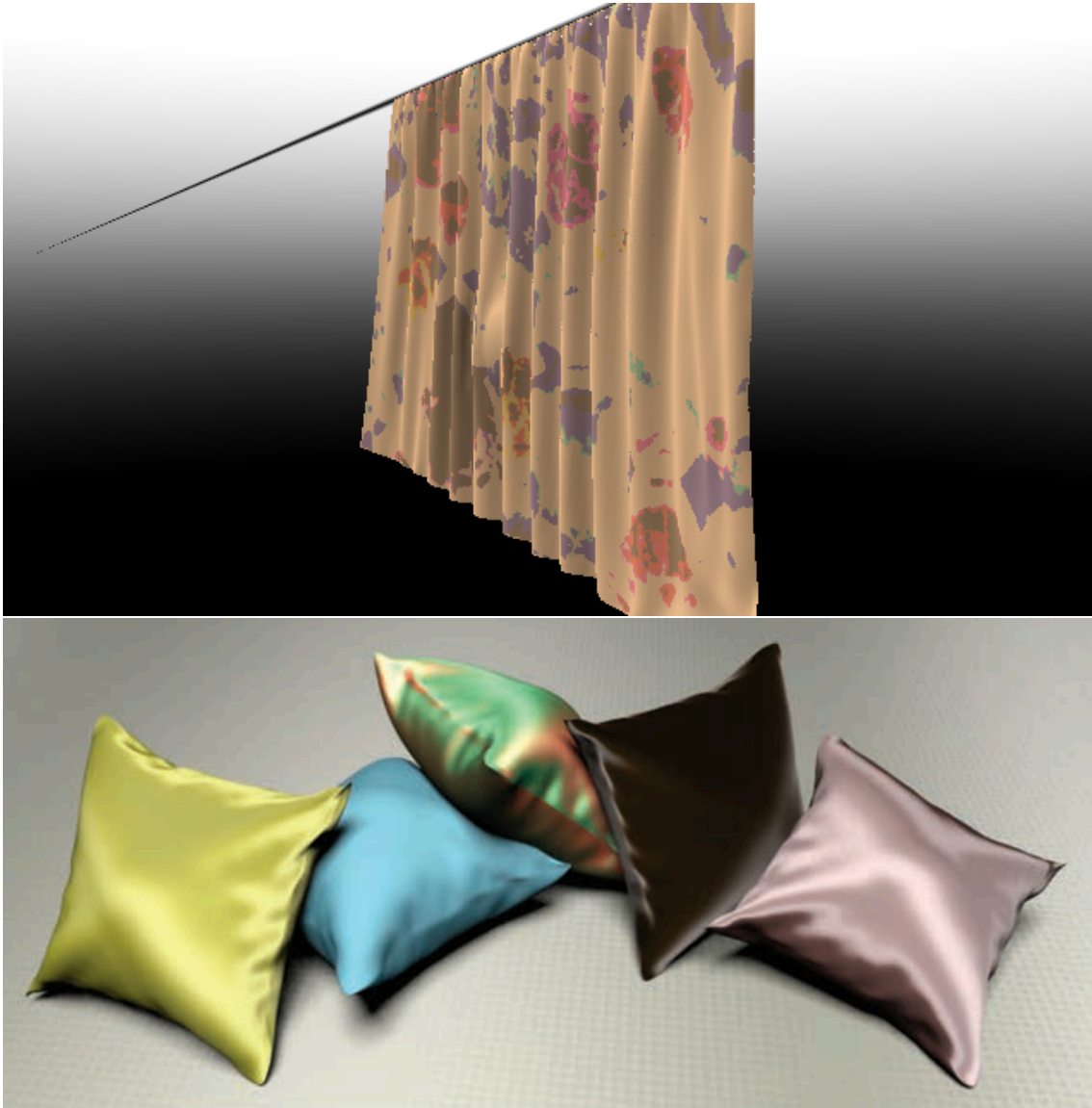
### 2.1.5 屏幕空间的环境光遮蔽 (难度等级: 2.0)



在作业中我们已经实现过了 Phong 光照模型。你可能还记得，环境项是一个常数，这看起来并不怎么有趣。

环境光遮蔽 (AO) 是增加环境项真实感的一种方法。(某些高级的方法甚至可以伪造全局光照!) 从理论上讲,“环境光遮挡”会在半球上均匀地发射阴影光线,以查看在一定距离内有多少光线被遮挡。但是,屏幕空间环境光遮蔽 (SSAO) 可以仅在图像平面上对此进行近似,达到实时效果,该技术广泛用于游戏中! 如果你选做本项目,我们要求你在 GPU 上使用着色器完成此工作,你可以使用 OpenGL / DirectX / Vulkan,当然也可以直接用我们课上提到的 ShaderToy,甚至可以使用 Unity / Unreal 引擎来完成!

### 2.1.6 布料模拟及布料材质的渲染 (难度等级: 2.0)



你可以扩展我们的绳子模拟小作业，仍使用质点弹簧系统，但让它能够支持布料的模拟。

对于布料的渲染我们知道，布料可以通过三种方式进行渲染：作为实际的纤维，作为参与的介质和作为表面。在这里，我们建议你使用[一个非常简单的表面模型](#)来渲染布料。

同样，由于单个任务过于简单，我们要求你在此项目中同时完成模拟与渲染这两部分任务。



## 2.2 渲染

### 2.2.1 体积散射 (难度等级: 2.0)



在作业中，你实现了对表面渲染，其中假定光线仅在撞击表面时才散射。在这种情况下，渲染方程式是场景中所有表面的积分。但是，这种表面渲染技术无法渲染一些体积散射效果，例如上面显示的雾。

要对体积散射建模，你需要计算场景中所有体积和所有表面的积分。你可以通过修改路径追踪器来做到这一点。主要区别在于，射线在撞击任何体积的表面之前可能会发生散射。你可以阅读以下两篇文章。

[Wojciech Jarosz](#) 的[这一章](#)介绍了有关体积散射的一些基本概念。[Lafortune](#) 和 [Willems](#) 撰写的[这篇论文](#)中有实现细节。你可以跳过本文的第 4 节，以获得简单的路径追踪实现。

### 2.2.2 渐进式光子映射 (难度等级: 2.5)



你可能已经注意到，路径追踪渲染器在渲染 caustics 效果时效率很低。为了改善这一点，你可以实现一种称为光子映射 (photon mapping) 的技术。光子映射在 caustics 上更有效，因为它允许路径样本在像素之间共享。光子映射的核心部分是一个能在场景中的某个点的邻域内查找光子的结构。可供选择的结构包括 KD-tree 和哈希网格。[PBRT 中的第 16.2 章](#)是一个很好的参考。

但是，光子映射是一种有偏的方法！直观地讲，由于阴影点使用的是其附近的光子，因此阴影效果总是模糊的。你需要查阅有关[渐进式光子映射的文章](#)以进行改进。

### 2.2.3 次表面散射 (难度等级: 2.0-2.5)



**BRDF**



**BSSRDF**

[这是一篇](#) 2001 年使用了 dipole approximation 的开创性的论文。较新的方法对此进行了很大的改进，但这是一个很好的起点。这有很多复杂的数学运算，但是一旦你理解了它要做什么，它的实现会比较简单。此外，这篇论文的内容非常偏理论，所以我们建议你自行搜索实现相关的文章与教程，而不是直接参考这篇论文的内容。

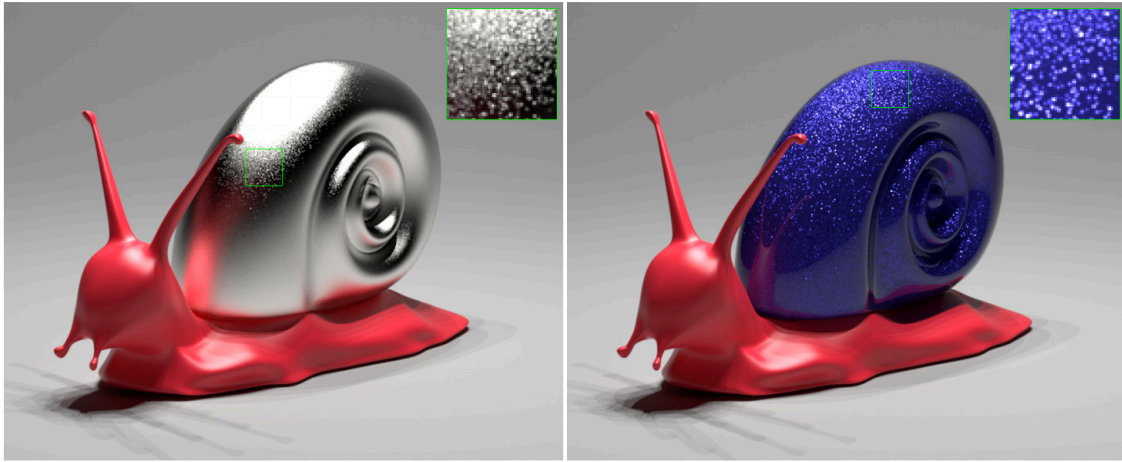
#### 2.2.4 头发渲染 (难度等级: 2.5)



你可以查阅[Marschner 等人的论文](#)。此外，[这篇论文](#)也是一个不错的参考。如果你打算实现头发的效果，你将会面临一个难题，光线与头发曲线的相交。因此，我们建议你使用已经实现了光线-头发相交的 Mitsuba(0.5.0) 渲染器。

如果你想挑战自我，也欢迎你实现[闫令琪老师的高级毛发反射模型](#)（难度等级：3.0）。

### 2.2.5 渲染复杂表面的闪光 (Glints) 效果 (难度等级: 3.0)



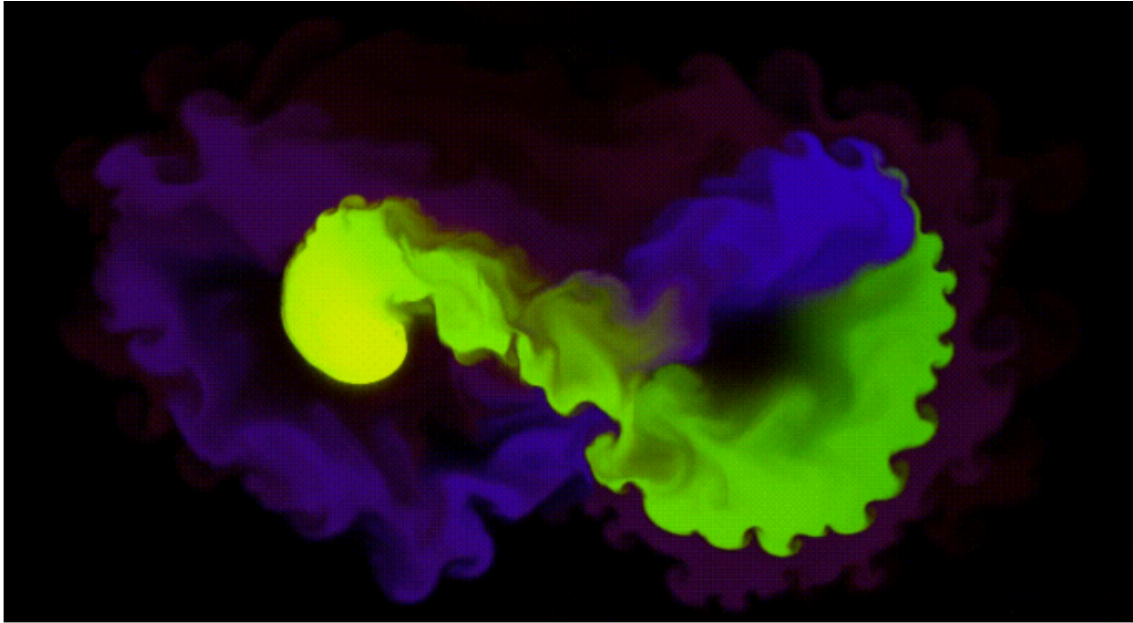
你也可以将纹理添加到路径追踪器中！之后，你可以扩展微表面模型以支持闪光效果！请查阅闫令琪老师的这两篇论文 ([论文一](#)和[论文二](#))，使用高分辨率法线贴图来实现闪光效果。你也可以在没有法线贴图的情况下实现[随机闪光模型](#)。

这里推荐一篇[微分光线](#)的论文 (在主光线旁边追踪稍微偏移的光线，使你可以计算一个像素在物体表面上覆盖的范围)。



## 2.3 动画

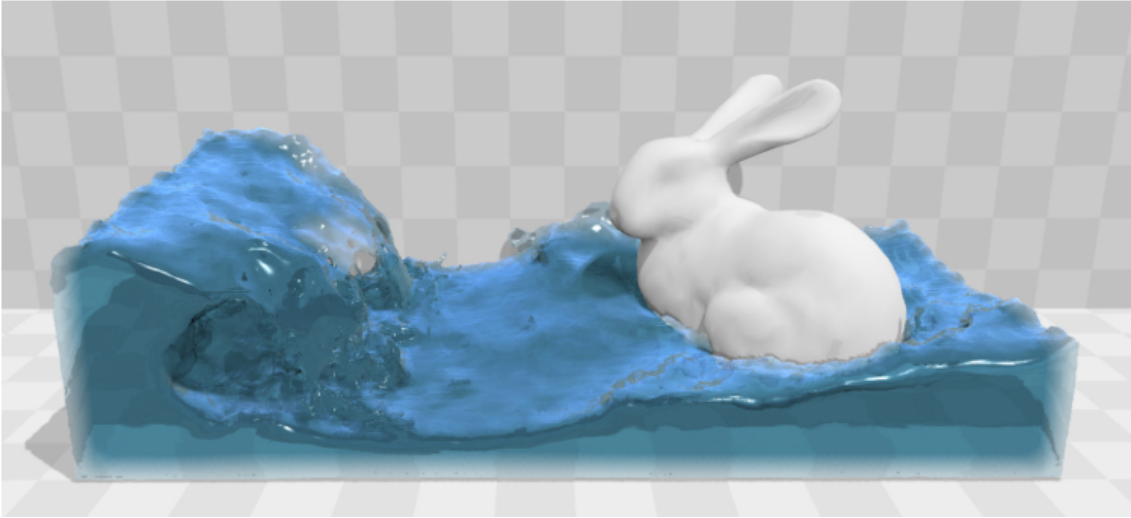
### 2.3.1 基于二维网格的流体动力学仿真 (难度等级: 2.0)



编写一个二维的流体动力学模拟程序，以估算Navier-Stokes 方程下不可压缩的均匀流体的行为。

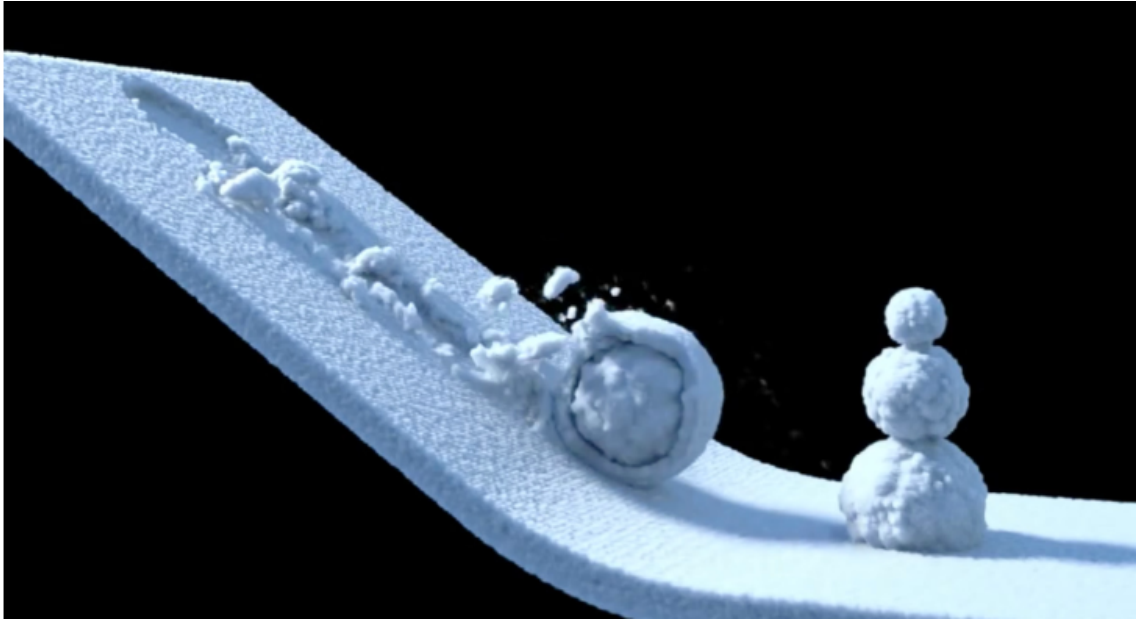
### 2.3.2 基于 3D 位置的流体模拟 (难度等级: 2.5)

$$\nabla_{p_k} C_i = \frac{1}{\rho_0} \begin{cases} \sum_j \nabla_{p_k} W(p_i - p_j, h) & \text{if } k = i \\ -\nabla_{p_k} W(p_i - p_j, h) & \text{if } k = j \end{cases} \quad (1)$$



3D 流体模拟可能听起来很吓人, 但请不用担心。[本文](#)介绍了一种相当简单的流体模拟算法, 将其作为多个球形粒子。从模拟的粒子中, 你可以构建表面并将其渲染为真实的水!

### 2.3.3 3D 雪景模拟 (难度等级: 3.0)



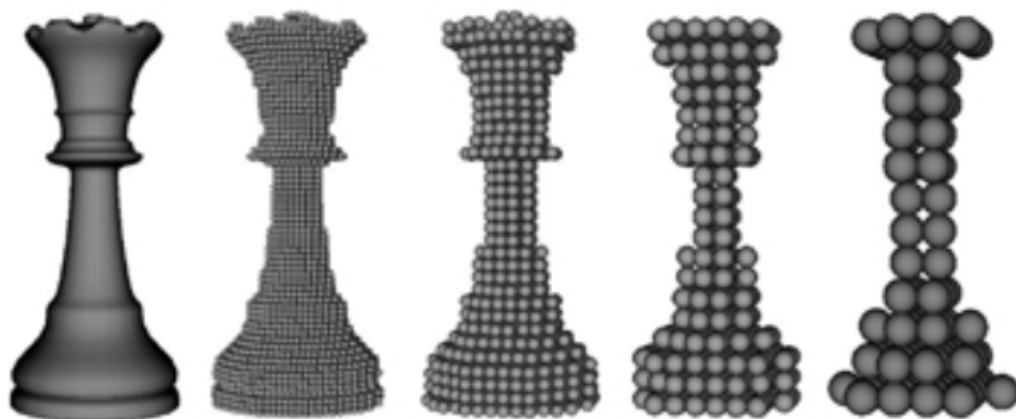
降雪是一种具有视觉挑战性的自然现象。[本文](#)介绍了一种基于粒子和网格的混合方法（物质点方法），用于各种参数的积雪模拟。

#### 2.3.4 3D 熔化模拟 (难度等级: 3.0)



想要新颖的方法来摧毁兔子吗？[本文](#)教你如何融化它。

### 2.3.5 刚体模拟 (难度等级: 2.5)



除了流体之外，我们还可以使用粒子来模拟刚体。这是一篇描述刚体动力学基础的[论文](#)。刚体仿真可以在 GPU 上实时完成，NVidia 的 GPU Gems 上有一篇很好的[文章](#)介绍了如何做到这一点。



## 2.4 其他

### 2.4.1 使用 NVIDIA OptiX 的 GPU 路径跟踪器 (难度等级: 1.5)



NVIDIA OptiX 是在 CUDA 之上构建的一组很酷的 API, 可以进行 GPU 光线追踪。它本身支持高效的 BVH 构造和光线与三角形相交。即使你使用的并不是 RTX 系列的 GPU, 你仍然可以期望运行速度提高 5 到 10 倍!

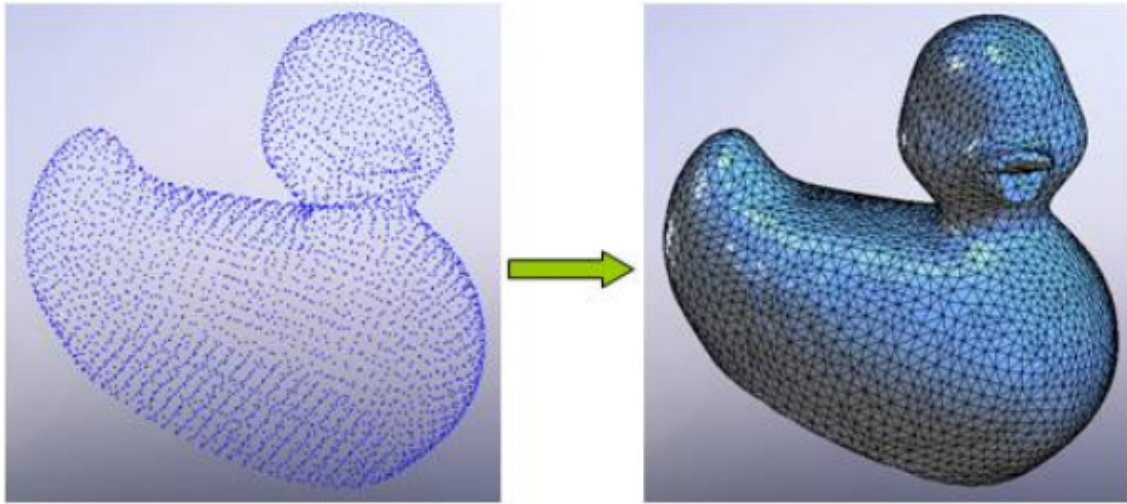
在此项目中, 你将会使用 OptiX 在 GPU 上编写路径追踪器。由于 OptiX 自己已经有基本的路径追踪实现, 因此在达到作业 7 的效果基础上, 我们还要求你实现最简单的 Cook-Torrance 微表面模型与环境光照。

#### 2.4.2 用两个三角形渲染世界（难度等级：2.5）



你需要多少个三角形来表示此场景并进行渲染？仅仅只需两个！[本文](#)将教你如何使用距离场表示几何图形，以及如何在片段着色器中实时渲染它们！注意，此项目要求你使用 GPU 的着色器，同样，你可以使用 OpenGL / DirectX / Vulkan，也可以直接使用 ShaderToy 来完成。

### 2.4.3 用点云来构建网格 (难度等级: 2.0)



该项目构想的目标是将通常由 3D 扫描仪获得的点云输入数据转换为网格表示。[本文](#)介绍了一种有趣且易于理解的算法，该算法运行良好。如果你想要更多挑战，则可以尝试实现[本文](#)。为了测试实现，你可以从该[存储库](#)中找到一些 3D 网格模型。然后，你可以在输入模型的顶点上运行网格重建算法。重建网格后，可以轻松将其与模型中的原始网格进行比较。

### 3 提交

- 首先请清理你的项目，并提交你的代码；
- 同时，请添加一个 `README.md` 文件，写清楚自己完成的项目是什么，并简述是如何实现的；
- 再请附上一些结果图片与演示视频，并在 `README.md` 中附上视频的链接（百度云等）；
- 最后，将上述内容打包，并用“姓名\_Final\_Project.zip”的命名方式提交到 SmartChair 平台。