作业 4: Kulla-Conty BRDF
GAMES202, 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 6 月 29 日 (星期一) 上午 10:00

截止时间为北京时间 2021 年 7 月 13 日 (星期一) 下午 8:00

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/games202/`
- 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 总览

微表面模型的 BRDF(Microfacet BRDF) 存在一个根本问题，就是忽略了微平面间的多次弹射，这就导致了材质的能量损失，并且当材质的粗糙度越高时，能量的损失会越严重。通过引入一个微表面 BRDF 的补偿项，来补偿光线的多次弹射，使得材质的渲染结果可以近似保持能量守恒，这个 BRDF 的补偿模型就是本轮作业需要重点完成的 Kulla-Conty BRDF 近似模型。

完成 Kulla-Conty BRDF 模型，关键在于计算 BRDF 的补偿项 f_{ms}，而 f_{ms} 的计算需要 $E(\mu)$ 和 E_{avg} 两个前置变量。由此本轮作业将分为以下两个部分：

- 在离线端 (lut-gen 文件夹中) 预计算 $E(\mu)$ 和 E_{avg}。
- 在实时端 (homework4 文件夹中) 通过查询预计算数据构建 BRDF 的补偿项。

2 实验流程

2.1 微表面 BRDF

在正式开始之前，需要注意的是对于微表面 BRDF，在本次作业中统一选用以下列出的 F, D, G 三项近似模型:

$$f_r(\boldsymbol{i}, \boldsymbol{o}) = \frac{F(\boldsymbol{i}, \boldsymbol{h})G(\boldsymbol{i}, \boldsymbol{o}, \boldsymbol{h})D(\boldsymbol{h})}{4(\boldsymbol{n} \cdot \boldsymbol{i})(\boldsymbol{n} \cdot \boldsymbol{o})}$$

F 项，选用 Schlick 近似：

$$F = R_0 + (1 - R_0)(1 - cos\theta)^5$$

D 项，选用 GGX 法线分布：

$$D_{GGX}(\boldsymbol{h}) = \frac{\alpha^2}{\pi((\boldsymbol{n} \cdot \boldsymbol{h})^2((\alpha^2 - 1) + 1))^2}$$

$$\alpha = roughness^2$$

G 项，选用与 GGX 法线分布匹配的 Smith 模型：

$$G_{Smith}(\boldsymbol{i}, \boldsymbol{o}, \boldsymbol{h}) = G_{Schlick}(\boldsymbol{l}, \boldsymbol{h})G_{Schlick}(\boldsymbol{v}, \boldsymbol{h})$$

$$k = \frac{(roughness + 1)^2}{8}$$

$$G_{Schlick}(\boldsymbol{v}, \boldsymbol{n}) = \frac{\boldsymbol{n} \cdot \boldsymbol{v}}{\boldsymbol{n} \cdot \boldsymbol{v}(1-k) + k}$$

在离线端和实时端的相应位置,你会需要上述几个公式辅助完成微表面 BRDF 相关代码。

2.2 预计算部分

该小节的工作集中在离线端，我们需要完成的任务是计算 $E(\mu)$ 和 E_{avg} 并将结果保存在相应图片中，有:

$$E(\mu_o) = \int_0^{2\pi} \int_0^1 f_r(\mu_o, \mu_i, \phi) \mu_i d\mu_i d\phi$$

$$E_{avg} = 2 \int_0^1 E(\mu) \mu d\mu$$

其中 $E(\mu)$ 随粗糙度和角度两个变量变化，而 E_{avg} 仅随粗糙度变化。在实时端通过这两个预计算能量，我们可以计算出补充 BRDF 项 f_{ms}。下面将详细介绍这两个变量的预计算过程应如何完成。

2.2.1 预计算 $\mathbf{E}(\mu)$

主要任务是要计算 $E(\mu)$:

$$E(\mu_o) = \int_0^{2\pi} \int_0^1 f_r(\mu_o, \mu_i, \phi) \mu_i d\mu_i d\phi$$

通过 cos 加权在半球上采样入射光方向 $\boldsymbol{i}$，使用蒙特卡洛方法来求解这个积分。

本轮作业要求完成的相应部分位于 **Emu_MC.cpp** 中的第 85 行，需要在这里计算被积函数的采样值。

如果一切顺利，运行 lut-Emu-MC 后应该可以得到如图 1 所示的结果。

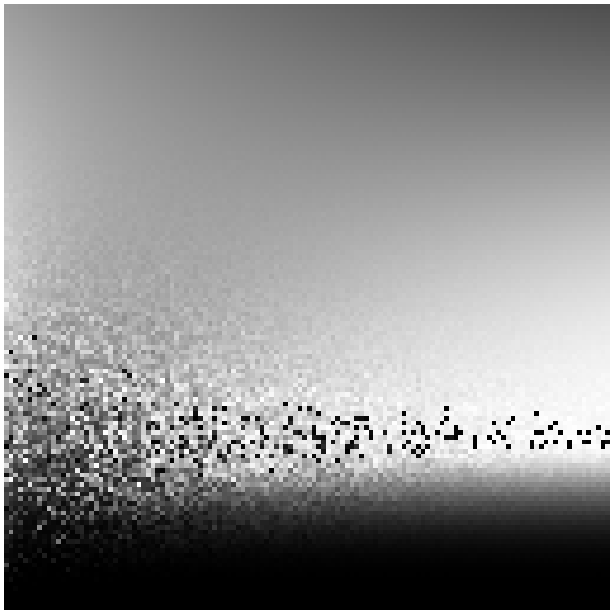

图 1: 预计算 E(μ)

可以看到在粗糙度较低时，我们计算出的积分值非常小并且有很多噪声。这是因为低粗糙度的微表面材质接近镜面反射材质，即微表面的法线 $\boldsymbol{m}$(即半程向量 $\boldsymbol{h}$) 集中分布在几何法线 $\boldsymbol{n}$ 附近，而我们由采样入射光方向 $\boldsymbol{i}$ 计算出的微表面法向量 $\boldsymbol{m}$ 分布并不会集中在几何法线 $\boldsymbol{n}$ 附近，也就是说这与实际低粗糙度的微表面法线分布相差很大，因此积分值的方差就会很大。实际上，我们可以通过对微表面法线 $\boldsymbol{m}$ 进行重要性采样来改善这个一问题，这部分属于本次作业的提高部分。

2.2.2 Bonus1: 通过重要性采样来预计算 E(μ)

重要性采样的方法有多种，这里我们介绍其中一种，你的实现也可以使用其他方法。我们将通过 GGX 采样来完成 $E(\mu)$ 的预计算工作。先从理论上讨论 GGX 采样算法，对于给定出射方向 $\boldsymbol{o}$ 的 GGX 采样，目标是采样生成入射方向 $\boldsymbol{i}$ 以计算 $\frac{f_r(\boldsymbol{i},\boldsymbol{o},\boldsymbol{h})(\boldsymbol{i},\boldsymbol{n})}{pdf_i(\boldsymbol{i})}$。因此，对于 GGX 算法有两个核心问题需要解决：如何采样和对应的概率 *pdf* 是什么。

第一个问题，如何采样入射方向 $\boldsymbol{i}$。我们首先根据选用的 NDF 模型，重要性采样微表面法向 $\boldsymbol{m}$（也就是 $\boldsymbol{i},\boldsymbol{o}$ 之间的半程向量 $\boldsymbol{h}$），随后通过采样得到的微表

面法向 $\boldsymbol{m}$，利用反射关系来计算入射方向 $\boldsymbol{i}$:

$$\boldsymbol{i} = 2(\boldsymbol{m} \cdot \boldsymbol{o})\boldsymbol{m} - \boldsymbol{o}$$

同时对于任意 NDF 下，采样 $\boldsymbol{m}$ 对应的概率密度 $pdf_m(\boldsymbol{m})$，有:

$$pdf_m(\boldsymbol{m}) = D(\boldsymbol{m})(\boldsymbol{m} \cdot \boldsymbol{n})$$

(这是由 NDF 中 $\int D(\boldsymbol{m})(\boldsymbol{m} \cdot \boldsymbol{n}) = 1$ 这一性质得出，文档不会涉及到关于 NDF 性质的讨论。同样，关于下述 GGX 采样点生成的推导过程也会被略去)

通过该 $pdf_m(\boldsymbol{m})$，可以计算出 GGX NDF 对应的采样点应该为:

$$\theta_m = arctan(\frac{\alpha\sqrt{\xi_1}}{\sqrt{1-\xi_1}})$$

$$\phi_h = 2\pi\xi_2$$

其中，$\xi_1, \xi_2 \in [0, 1)$。

第二个问题，如何计算采样得到的入射方向的概率。因为我们最后生成的采样方向是入射方向 $\boldsymbol{i}$, 所以最后结果的权重应该是:

$$weight(\boldsymbol{i}) = \frac{f_r(\boldsymbol{i}, \boldsymbol{o}, \boldsymbol{h})(\boldsymbol{i}, \boldsymbol{n})}{pdf_i(\boldsymbol{i})}$$

所以需要将之前采样微表面法线的概率密度 $pdf_m(\boldsymbol{m})$ 转换成采样入射的概率密度 $pdf_i(\boldsymbol{i})$，而两者之间的转换只需要简单的乘一个 **Jacobian** 项即可，即:

$$pdf_i(\boldsymbol{i}) = pdf_m(\boldsymbol{m}) \parallel \frac{\partial \omega_m}{\partial \omega_i} \parallel$$

其中，对于反射有:

$$\parallel \frac{\partial \omega_m}{\partial \omega_i} \parallel = \frac{1}{4(\boldsymbol{i} \cdot \boldsymbol{m})}$$

讨论完以上两点后，最终对于采样入射的权重可以整理为:

$$weight(\boldsymbol{i}) = \frac{(\boldsymbol{o} \cdot \boldsymbol{m})G(\boldsymbol{i}, \boldsymbol{o}, \boldsymbol{h})}{(\boldsymbol{o} \cdot \boldsymbol{n})(\boldsymbol{m} \cdot \boldsymbol{n})}$$

接下来正式进入代码实现，在 **Emu_IS.cpp** 中有三个任务点需要完成 (总计代码量为 20-30 行):

- GGX 的采样工作（第 26 行开始）
- 微表面模型中 G 项 Smith 近似模型的补充工作（第 44 行开始）
- 计算最后返回的权重（第 70 行开始）

值得注意到一点是，为了方便实时端读取，向图片中写入数据时可以翻转一下，使原点转移到左下角（默认原点在左上角）。如果一切顺利，运行程序后你应该可以得到和下列一样的预计算图片：

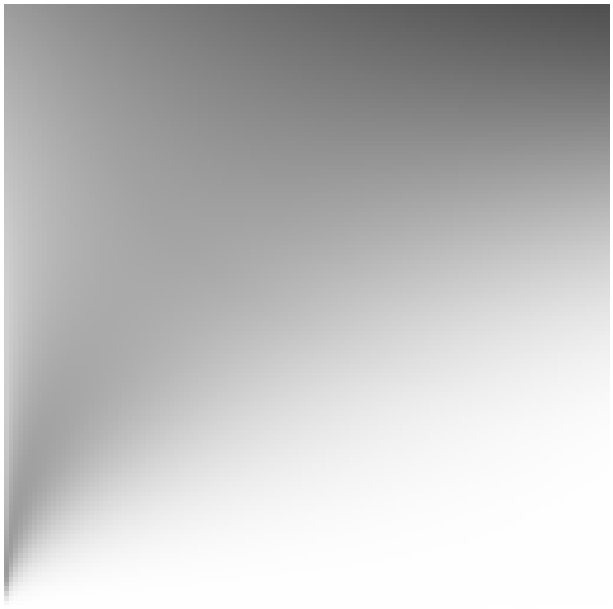

图 2: 预计算 E(μ)

进一步验证，如果保存的预计算结果是 $1 - E(\mu)$，并且关闭写入翻转（空框架的第 96 行）你可以得到的和本节课对应 PPT 中一样的预计算结果（PPT 第 28 页）：

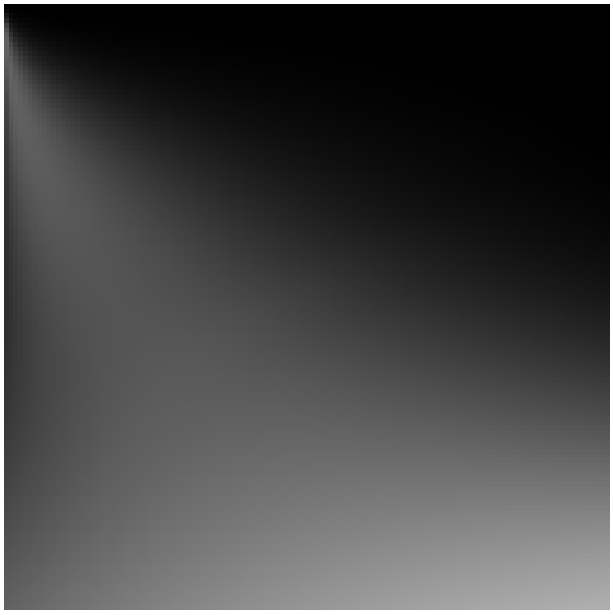

图 3: 关闭写入翻转，预计算 1.0-E(μ)

2.2.3 预计算 $\mathbf{E}_{avg}$

接下来对于 E_{avg} 的预计算就非常简单了，预计算的 E_{avg} 应该只随 $roughness$ 变化而变化。所以我们采用的方案是，首先对于之前预计算得到的 $E(\mu)$ 同理采样出射，计算出对应的 $E(\mu)\mu$，随后按 $roughness$ 将一行的 $E(\mu)\mu$ 累加除以行宽即可，这样可以认为通过简单的均匀采样完成了 E_{avg} 积分的近似计算。需要补充完成的代码位于 **Eavg_MC.cpp** 的第 81 行，需要计算采样值在下列积分式内被积函数的值：

$$E_{avg} = 2\int_0^1 E(\mu)\mu d\mu$$

如果一切顺利，你可以得到诸如下图所示的预计算 E_{avg} 图片。可以注意到对于我们的保存方式来说，粗糙度随行从下到上增加，而同一行上的存储值应该是相同的：

图 4: 预计算 E_{avg}

E_{avg} 表示我们能够看到的能量，在粗糙度低时微表面能量损失少，我们看到的能量多存储结果偏白；在粗糙度高时能量损失多，我们看到的能量少存储结果偏黑，整体上和我们的认知是一致的。

2.3 实时端计算补充 BRDF

接下来的工作转移到了实时端作业框架中，使用与预计算的数据计算模拟经过多次反射后出射的补充 BRDF。该补充 BRDF 与微表面 BRDF 相加即可构成最终我们需要的 Multiple Scattering BRDF。

我们使用了两排 Shading Ball 进行对比展示，上一排是微表面 BRDF+ 补充 BRDF，下一排是微表面 BRDF，从右到左粗糙度逐渐升高。我们期待的结果是：当粗糙度高时，微表面 BRDF 的能量损失会更严重此时渲染结果会偏暗，与上一排有补充 BRDF 的 Shading Ball 相比没有那么明亮；而在粗糙度低的时候，上下两排对应的 Shading Ball 渲染结果应该很相似（注意，对于场景中的 Shading Ball 实际上仅被一个方向光源照亮）。

根据上述展示方式，有如下两个任务点需要完成：

- 微表面材质对应 Shader 的补充完成

- Kulla-Conty 材质对应 Shader 的补充完成

2.3.1 微表面模型

本轮工作主要集中在 **pbrShader/PBRFragment.glsl** 中，需要根据2.1小节列出的微表面 BRDF 中 F、G、D 的近似模型公式补充完成代码。

在完成本小节任务时，可以在 **engine.js** 中先注释掉上一排的 Shading Ball (第 99-108 行代码)，以此来运行查看 PBR 材质是否编写正确。

如果一切顺利，应该可以看到如下图所示的使用微表面 BRDF 材质的一排 Shading Ball：

图 5: 使用微表面 BRDF 的材质

可以看到越靠近左侧也就是粗糙度越高的 Shading Ball 颜色越暗，这意味着能量损失越严重，接下来我们将通过 Kulla-Conty 近似模型，引入一个补充 BRDF 将单次反射的 BRDF 升格为多次反射的 BRDF。

2.3.2 Kulla-Conty 模型

本轮工作将集中在 **KullaContyShader/KullaContyFragment.glsl** 中，根据之前预计算的 $E(\mu)$ 以及 E_{avg}，我们可以计算出补充 BRDF 项 $f_{ms}(\mu_o, \mu_i)$：

$$f_{ms}(\mu_o, \mu_i) = \frac{(1 - E(\mu_o))(1 - E(\mu_i))}{\pi(1 - E_{avg})}$$

同时对于有颜色的材质，会有自然存在的能量损失，这样就需要在之前计算得到的 $f_{ms}(\mu_o, \mu_i)$ 上乘上一个附加的 BRDF 项 f_{add}：

$$f_{add} = \frac{F_{avg}E_{avg}}{1 - F_{avg}(1 - E_{avg})}$$

其中，$F_{avg} = 2\int_0^1 F(\mu)\mu d\mu$

关于 F_{avg} 的计算，我们引用了Revisiting Physically Based Shading at Imageworks第 26 页给出的方式，误差保持在可接受范围内。

最后对于 Kulla-Conty 材质，可以得到的 BRDF 项 f_r 为：

$$f_r = f_{micro} + f_{add} * f_{ms}$$

本部分需要完成的代码主要集中在第 70 行开始。如果一切顺利此时运行程序应该可以看到上一排的 Kulla-Conty 材质会的 Shading Ball 会比下一行的微表面材质在粗糙度高时表现明显更好，而在粗糙度低时两者的表现会比较接近。

值得注意到是，对于基础部分我们不建议使用过于低粗糙度的 Shading Ball 进行对比。因为 cos 权重采样在低粗糙度的时候误差非常大，可以注释掉 **engine.js** 中两个粗糙度为 0.15 的小球后比较观察。如果完成了使用重要性采样进行预计算的方案，那么就不用担心低粗糙度的问题了。

图 6: 使用 Kulla-Conty BRDF 补充微表面模型能量损失的材质

3 评分与提交

提交时需要删除/build /lib, /assets 等所有与你实现代码无关的文件夹，并在建立的 /images 文件夹中保存运行截图。

/images 文件夹中需要包括预计算的 $E(\mu)$，E_{avg} 以及实时端截图。同时在 README.md 中简要描述本轮工作，尤其是完成了作业的哪些部分。如果完成的任务中包含了提高部分请注明改动了哪些代码、文件以方便助教同学批改。

3.1 评分

- [5 分] 实现预计算 $E(\mu)$。
- [5 分] 实现预计算 E_{avg}。
- [10 分] 正确实现 PBR 材质。
- [5 分] 正确实现 Kulla-Conty 材质。
- [10 分] Bonus 1: 实现重要性采样的预计算方法。
- [5 分] Bonus 2: 在预计算 $E(\mu)$ 时，使用 Split Sum 完成预计算工作。

- [-3 分] 惩罚分数:
  未删除 /lib, /assets, assignment4.pdf 等与代码无关的文件夹。
  未按格式建立 /images, 缺少结果图片。

3.2 提交

作业提交使用的平台为 **Smartchair** 平台，地址为http://www.smartchair.org/GAMES202/。