Real-Time Environment Mapping

Mengzhu Wang

2024 年 5 月 13 日

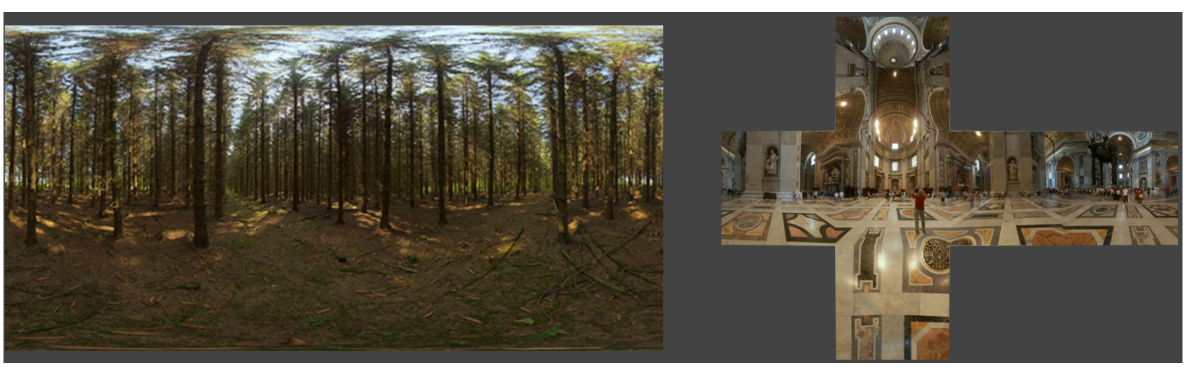
1 Environment Lighting

环境光照又称环境贴图（Environment Map），记录场景中任意一点在不同方向上接收到的光照强度。它认为环境光照都来自无穷远处，光照强度在场景中所有位置都是均匀的，因此使用环境贴图渲染会产生漂浮感。

基于图像的光照（Image Based Lighting, IBL）是指一类通过环境贴图保存环境光信息，从而实现基于物理的渲染方法。IBL 通常使用 Spherical Map 和 Cube Map。

Spherical Map 是指将一个理想高反射的球体放在场景中，球体表面记录了各个方向的光线信息，展开球面后得到一张长方形全景图。但是纬度高的地方在球面上占据面积小，球面无法均匀展开为平面，上下边界会发生扭曲。

Cube Map 是指用立方体包住记录环境光的单位球面，从球面向立方体的 6 个面分别射出光线，渲染到立方体面上，其展开不存在扭曲的问题。



2 Shading from Environment Lighting

2.1 Problem Setting

使用环境光贴图实现 IBL，本质上是对 shading point 的渲染方程的求解。不考虑阴影所以去掉 visibility 项，lighting 项是环境光照，其式如下：

$$L_o(p,\omega_o) = \int_{\Omega^+} L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)\cos\theta_i d\omega_i$$

求解以上方程的常用方法是蒙特卡洛积分，但是它需要大量的采样才能比较准确，导致速度很慢。避免采样是解决实时性问题的关键，IBL 中最为常见的算法是 The Split Sum Approximation。

2.2 The Split Sum Approximation

当 BRDF 是 glossy 时，其 lobe 是花瓣状的，只有很小的积分域接受环境光，即有一个小的支撑集。当 BRDF 是 diffuse 的，其 lobe 是均匀的半球状，对于任意方向的环境光变化都不大，即是光滑的。这两种情况都满足如下近似公式的条件

$$\int_{\Omega} f(x)g(x)dx \approx \frac{\int_{\Omega_G} f(x)dx}{\int_{\Omega_G} dx} \cdot \int_{\Omega} g(x)dx$$

根据上述近似公式，渲染方程可以拆为 lighting 积分和 BRDF 积分两部分，即

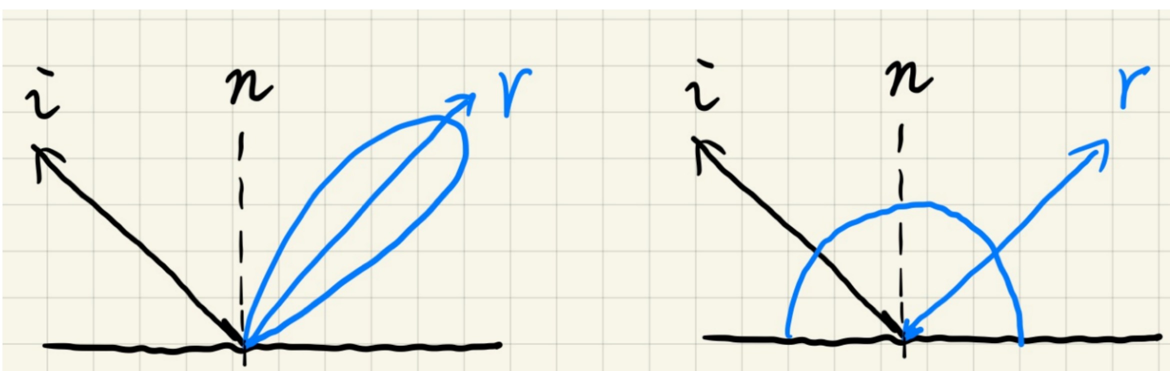


图 1: left: glossy; right: diffuse

$$L_o(p.\omega_o) \approx \frac{\int_{\Omega_{f_r}} L_i(p,\omega_i)d\omega_i}{\int_{\Omega_{f_r}} d\omega_i} \cdot \int_{\Omega^+} f_r(p,\omega_i,\omega_o)\cos\theta_i d\omega_i$$

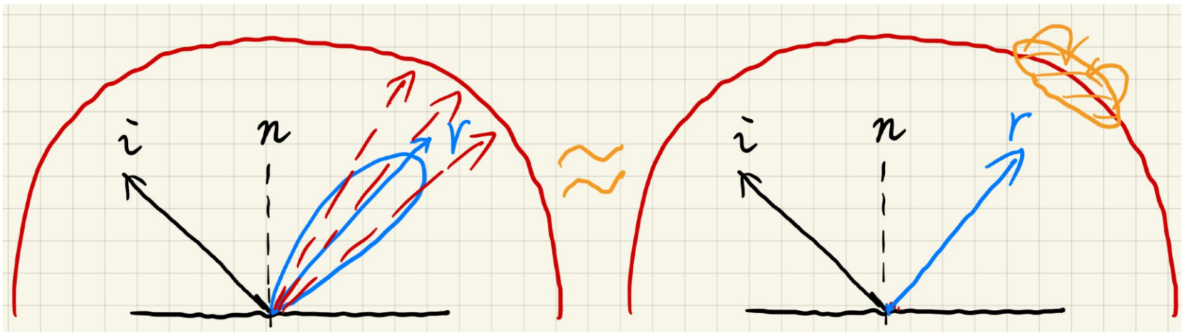
这两个部分都可以通过预计算减少开销，无需进行多次采样。

2.2.1 1st Stage: Prefiltering of the environment lighting

lighting 积分的表达式是

$$\frac{\int_{\Omega_{f_r}} L_i(p,\omega_i)d\omega_i}{\int_{\Omega_{f_r}} d\omega_i}$$

其表示把 BRDF 的 lobe 覆盖区域的 lighting 积分，然后取平均，类似一个模糊或者滤波的操作，滤波核的大小就是 BRDF 的 lobe 覆盖的区域。这部分预滤波的计算可以在渲染之前完成，只需要对滤波后的环境贴图采样一次光线方向就能得到积分，即先采样后滤波可以近似为先滤波后采样。



BRDF 的 lobe 形状越尖锐，即环境光积分范围越小，就需要使用模糊程度更低的环境贴图；反之 BRDF 的 lobe 形状越粗壮，即环境光积分范围越大，就需要使用模糊程度更高的环境贴图。对于这些不同大小的滤波核，可以参考 MIPMAP 的做法，生成不同 level 的环境贴图，通过三线性插值得到任何模糊程度的滤波结果。

2.2.2 2nd Stage: Precompute BRDF integral

Microfacet BRDF 的表达式是

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

其中 $F(i,h)$ 是菲涅尔项（Fresnel term），$G(i,o,h)$ 是几何项表示阴影遮挡，$D(h)$ 是法线分布（NDF）。

Fresnel term 可以用 Schlick's approximation 近似，其表达如下

$$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5$$
$$R_0 = (\frac{n_1 - n_2}{n_1 + n_2})^2$$

NDF term 为 Beckmann distribution 时可以用下式表达

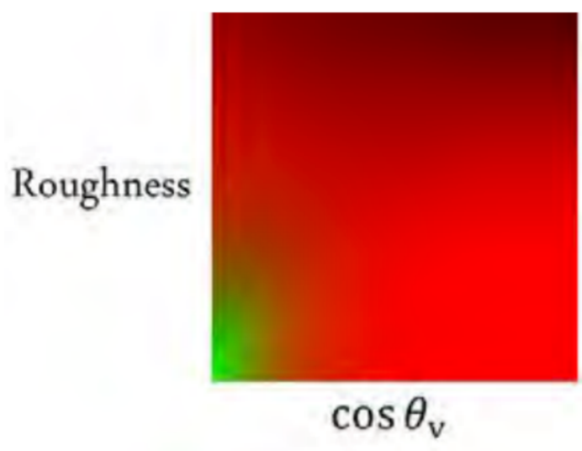
$$D(h) = \frac{e^{\frac{\tan^2\theta_h}{\alpha^2}}}{\pi\alpha^2\cos^4\theta_h}$$

Fresnel term 与基础反射率 R_0 以及入射角有关，NDF term 与粗糙度 α 以及半角与法线的夹角 θ_h 有关。由于入射角、反射角 θ_v 和半角相差不大，可以认为是一个角，而 θ_h 可以通过它们得到，因此把角度当作一个维度。BRDF 此时共有三个维度。

将 Fresnel trem 代入到 BRDF 积分项中可得

$$
\begin{aligned}
&\int_{\Omega+} f_r(p,\omega_i,\omega_o)\cos\theta_i d\omega_i \\
=&\int_{\Omega+} \frac{f_r(p,\omega_i,\omega_o)}{F} F\cos\theta_i d\omega_i \\
\approx&\int_{\Omega+} \frac{f_r(p,\omega_i,\omega_o)}{F}(R_0+(1-R_0)(1-\cos\theta_i)^5)\cos\theta_i d\omega_i \\
=&R_0\int_{\Omega+}\frac{f_r}{F}(1-(1-\cos\theta_i)^5)\cos\theta_i d\omega_i+\int_{\Omega+}\frac{f_r}{F}(1-\cos\theta_i)^5)\cos\theta_i d\omega_i
\end{aligned}
$$

R_0 被拆分出来，并把 BRDF 积分拆成两个积分，它们都只依赖于 α 和 θ_v，所以可以针对这两项进行预计算，建立二维查询表。该过程只适用于各向同性的 BRDF。



2.2.3 Plus

在工业界中，通常将积分转换为求和

$$
\frac{1}{N}\sum_{k=1}^{N}\frac{L_i(l_k)f(l_k,v)\cos\theta_{l_k}}{p(l_k,v)} \approx (\frac{1}{N}\sum_{k=1}^{N}L_i(l_k))(\frac{1}{N}\sum_{k=1}^{N}\frac{f(l_k,v)\cos\theta_{l_k}}{p(l_k,v)})
$$

3 Shadow from Environment Lighting

实时渲染中实现环境光照阴影十分困难。

- many-light problem：环境光可以认为是很多小光源组成的大光源，即转换为 many-light 问题。用环境光生成阴影也就是说每个小光源都要生成一张 shaodow map，其消耗是线性于光源数量的。
- sampling problem：环境光渲染本质上是求解渲染方程，可以通过采样方法。对于每个 shading point 来说不知道遮挡关系，而且无法保证 small support 和 smooth，很难把 visibility 项从渲染方程中提取出来。

工业界的解决方案是选择最主要的光源生成一个或几个阴影。

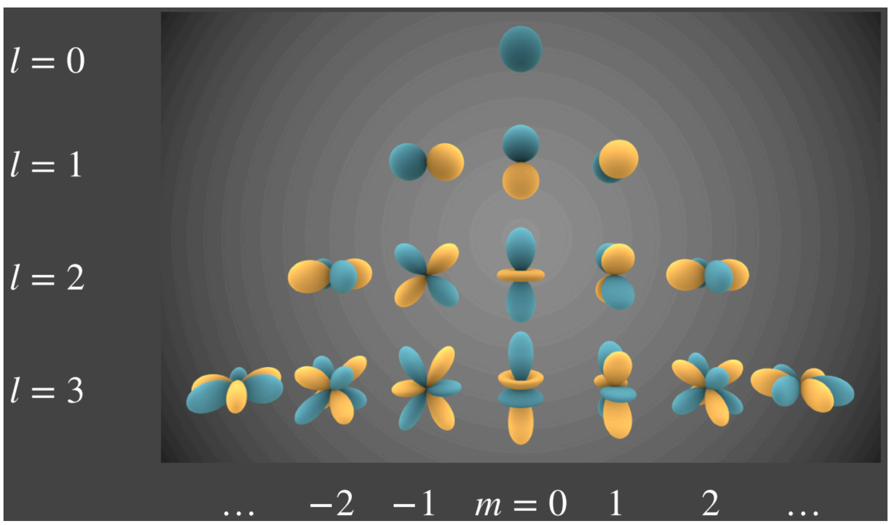
生成环境光下的阴影还有几个其他的方法

- Imperfect shadow maps
- Light cuts
- RTRT (real-time ray tracing)
- Precomputed radiance transfer (PRT)

4 Real-time Environment Lighting

4.1 Spherical Harmonics

球谐函数是指定义在球面上的一系列二维基函数，每个基函数都可以用勒让德多项式（Legendre polynomial）表示。SH 在第 l 阶有 $2l+1$ 个基函数，前 n 阶共 n^2 个基函数。阶数越高，表示的频率就越高。



SH 的展开式为一系列基函数 $B_i(x)$ 的线性组合 $f(x) = \sum_i c_i \cdot B_i(x)$，其中 $B_i(x)$ 已知，只需要求出每个基函数的系数。SH 系数的计算过程称为投影，也就是 product integral，其式为 $c_i = \int_\Omega f(\omega)B_i(\omega)d\omega$。重建过程中使用的基函数越多，展开的阶数越高，还原的效果就越好。

4.2 Prefiltered env. lighting

Diffuse 情况下，在渲染方程中计算 lighting 乘以 BRDF（忽略 visibility 和 cos），lighting 项是环境光贴图的球面函数，BRDF 是定义在半球上的光滑函数，这两部分逐点相乘并积分即 $\int_\Omega f(x)g(x)dx$，这个操作对应 product integral 的定义。product integral 相当于卷积或滤波的操作，结果的频率是由积分项中最低频的函数决定的。由于 diffuse 的 BRDF 低频，lighting 也无需使用高频的环境光贴图，前三阶 SH 近似 lighting 就能得到较好的结果。

4.3 Precomputed Radiance Transfer (PRT)

考虑 visibility，环境光渲染方程为

$$L(o) = \int_{\Omega} L(i)V(i)\rho(i,o)max(0, n \cdot i)di$$

其中 lighting 项、visibility 项和 BRDF 项都可以描述为球面函数，然后逐像素相乘再相加。这个过程计算量很大，速度很慢。

PRT 的思路是将渲染方程分为 lighting 和 lighting transport 两部分，假设场景中只有 lighting 会变化，而 lighting transport 不变。lighting 部分用 SH 展开，近似为基函数的线性组合，在预计算时需要计算出 light transport，投影到基函数空间。

4.3.1 Diffuse Case

a)

Diffuse 情况下，BRDF 可以认为是一个常数，提取到积分外面，即

$$L(o) = \rho \int_{\Omega} L(i)V(i)max(0, n \cdot i)di$$

其中的 lighting 可以用基函数组合 $L(i) \approx \sum l_i B_i(i)$ 近似，那么渲染方程近似为

$$L(o) \approx \rho \sum l_i \int_{\Omega} B_i(i)V(i)max(0, n \cdot i)di = \rho \sum l_i T_i$$

light transport 是固定的，基函数 B_i 也是固定的，T_i 就是将 light transport 投影到基函数上，T_i 的另一种理解是将 B_i 当作是光照来渲染场景，它可以预计算得到。最后渲染方程就变为点积运算，这里包含了 visibility，因此结果是有阴影的。

在这个过程中，预计算 light transport 意味着对于场景中任意一个 shading point，看到的 light 对其他东西的遮挡是一样的，即场景不能动。而 lighting 是可以变化的。

基函数选用 SH 函数，它具有如下性质

- 正交性
- 投影计算简单
- 旋转原函数相当于旋转该函数对应的所有基函数，旋转后的基函数可以被同阶的 SH 函数线性表示（解决光源旋转问题）

b)

分别将 lighting 和 light transport 预计算，各自用 SH 展开

$$
\begin{aligned}
L(\omega_i) &\approx \sum_p c_p B_p(\omega_i) \\
T(\omega_i) &\approx \sum_q c_q B_q(\omega_i) \\
L_o(p, \omega_o) &= \int_{\Omega+} L_i(p, \omega_i) f_r(p, \omega_i, \omega_o) \cos\theta_i V(p, \omega_i) d\omega_i \\
&= \sum_p \sum_q c_p c_q \int_{\Omega+} B_p(\omega_i) B_q(\omega_i) d\omega_i
\end{aligned}
$$

渲染方程变为双重求和，其中的积分表示将任意一个基函数投影到基函数上。由于 SH 的正交性，只有 p=q 时积分项才为 1，其余为 0，相当于一个二维矩阵只有对角线上有值，所以时间复杂度并非 $O(n^2)$，而只是 $O(n)$，与第一种点积计算相同。

4.3.2 Glossy Case

diffuse 的 BRDF 是常数，glossy 的 BRDF 是是四维的，包括入射角、出射角、入射方位角、出射方位角。类似 diffuse case 的第一种计算，将 light transport 投影到 lighting 的基函数上时，投影的结果不再是一个 SH 系数，而是一个与出射方向 o 相关的函数 $T_i(o)$，渲染方程可以写成如下形式

$$L(o) = \int_\Omega L(i)V(i)\rho(i,o)max(0, n \cdot i)di \approx \sum l_i T_i(o)$$

把 $T_i(o)$ 球谐展开为 $T_i(o) \approx t_{ij} B_j(o)$，代入上式得到

$$L(o) \approx \sum (\sum l_i t_{ij}) B_j(o)$$

其中 t_{ij} 是 transport matrix，$l_i t_{ij}$ 也就是向量 light coefficient 和 transport matrix 相乘得到向量 reflected radiance coefficient。最后的渲染计算实际上是向量矩阵相乘。

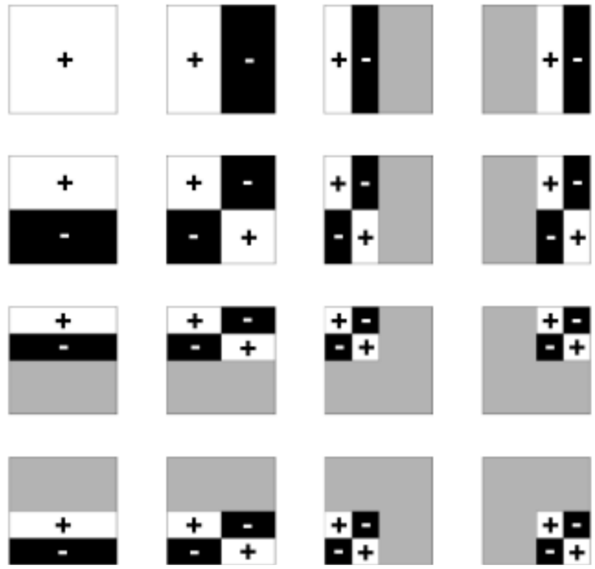
通常使用 9/16/25 的 SH 基函数，glossy 渲染计算量比 diffuse 大。对于接近镜面反射的情况，SH 并不适用。

4.3.3 Limitations

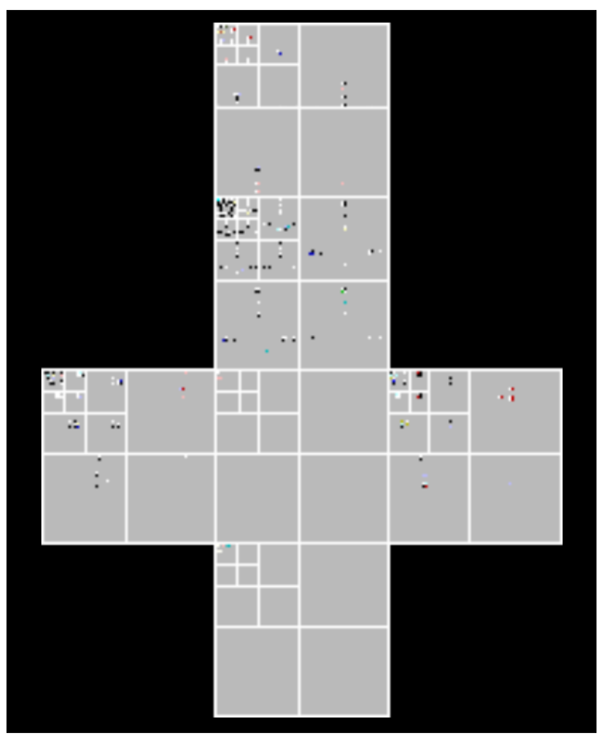
- 实际中 SH 只适合描述低频
- 动态 lighting，静态场景
- 大量预计算数据

4.4 Wavelet

小波函数定义在图像上，不同小波函数有不同的定义域。将原函数投影到全部基函数上，只有一小部分小波系数非零，只保留这部分用于恢复原函数。它能够支持全频率的表示。



对于小波函数作为基函数的情况，使用 cube map 记录环境光照。小波变换将每个图像分成四块，左上是低频信息，其他部分是高频信息，左上继续进行小波变换，依此类推。最后只剩下非零的主要信息，实现大幅压缩。



小波函数解决了全频率的问题，但其不支持光照旋转。