基于蒙特卡洛二维空间上的光线传输模拟

内容摘要： 自计算机图形学发展至今，已经有了四十多年的历史。目前对于三维空间中光线与表面或介质交互的模拟已经非常接近于真实的物理现象，例如反射、折射、次表层散射、体积散射等。几种主要的算法有光线追踪、光子映射、辐射度算法。本文受到了在三维空间中路径追踪算法的启发，提供了一种在二维空间上进行光传输模拟的思路，包括求交方法、采样方法、光与表面的交互、焦散等。

关键词： 光线求交；蒙特卡洛方法；路径追踪

0 绪论

传统的光线追踪（Ray Tracing）算法是：追踪从摄像机发射的光线直到与某一表面相交，然后计算该点在场景中的辐射率，并最终将其转换为颜色信息反映在屏幕上的一个像素中，图一展示了光线追踪的基本思路。

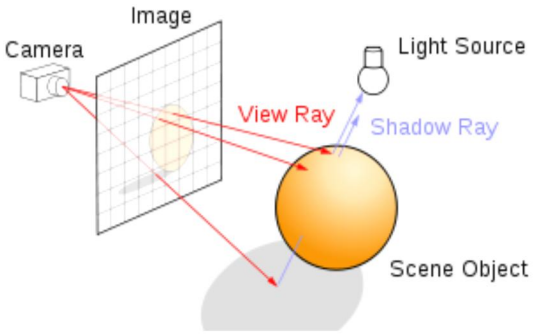


图 1 光线追踪算法

该算法基于了以下物理事实：即三维空间中某点的辐射率等于以该点为中心的上半球空间 $H^2(n)$，来自所有方向的辐射率之和。用数学语言描述即：

$$\mathrm{L_o}(\mathbf{p},\boldsymbol{\omega}_i)=\int_{H^2} f_r(\mathbf{p},\boldsymbol{\omega}_o,\boldsymbol{\omega}_i)L(\mathbf{p},\boldsymbol{\omega}_i)|\cos\theta_i|\,d\boldsymbol{\omega}$$

该方程又称为渲染方程（Rendering Equation）如图 2 所示。其中 $\mathrm{f}_r(p,\omega_o,\omega_i)$ 为 BSDF 项，描述了点 p 上一个方向对 (ω_o,ω_i) 的入射与出射光的比例关系。

显然这是一个高维积分，通常没有解析解，因此需要借助蒙特卡洛积分对其求近似数值解。

蒙特卡洛方法的中心思想就是，以大数定律为理论依据，对某个分布进行采样（Sampling），后对样本求均值。当样本足够多时（理论值为无穷大），所得的平均值与真实值相等。

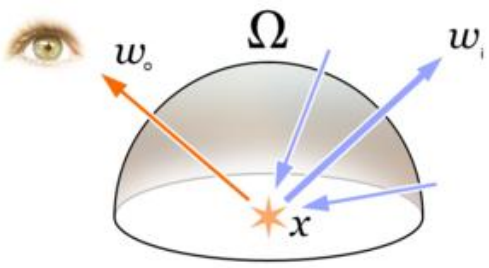


图 2 渲染方程图解

把这种思想应用到光线追踪算法中：追踪从视点发射的光线直到与某一物体相交，然后以该点为新的起点，由该点所在的表面的特性提供新的方向，如此再递归式地追踪下去直到遇到光源或从场景中逸出。

由于每从视点出发的一条光线经过与场景的交互（比如反射、折射）产生的多条光线组成了一条路径，故这种应用了蒙特卡洛方法的光线追踪算法又叫做路径追踪（Path Tracing），而其蒙特卡洛思想则体现在对单一像素进行多次采样，而每次采样在场景中的路径都不相同，最后求平均则可近似该像素的真实值。该过程可以表示为：

$$\mathrm{L}(x,y)=\frac{1}{\mathrm{N}}\sum_{i=1}^{N}\frac{f_r(\mathbf{p},\boldsymbol{\omega}_{\mathrm{o}},\boldsymbol{\omega}_i)L(\mathbf{p},\boldsymbol{\omega}_i)|\cos\theta_i|}{p(\theta)}$$

其中 N 为样本数， $\mathrm{p}(\theta)$ 为 θ 的概率密度函数(Probability Density Function)．

1 二维渲染方程

本文章以三维空间中路径追踪算法为基础，在二维空间中用蒙特卡洛方法近似求解渲染方程。

两者的不同之处在于，三维空间中所有的点最终都将映射到一个二维空间上（因为最终将呈现在屏幕上，屏幕是一个二维空间）。

而二维空间本身就是平面，故不需要像三维空间找到一个映射关系映射到二维空间，直接对每个点求解渲染方程即可。具体来说，对于一个点 $\mathrm{p}(x,y)$ 所接收到的光照就是以该点为中心，从 360° 每个微分方向 $\mathrm{d}\theta$ 接受到的光的总和。则二维空间的渲染方程如下：

$$\mathrm{L}(x,y)=\int_0^{2\pi}L(\mathbf{p},\theta)d\theta$$

其中 $\mathrm{L}(\mathrm{p},\theta)$ 表示点 $\mathrm{p}(x,y)$ 在 θ 方向接受到的光照。

与三维空间的渲染方程类似，上式也需要借助蒙特卡洛积分求解：

$$\mathrm{L}(x,y)=\frac{1}{\mathrm{N}}\sum_{i=1}^{N}\frac{L(\mathbf{p},\theta)}{p(\theta)}$$

2 光线求交

至此，已将所有关于光线在场景中的传输形式的基本内容介绍完毕，但还需要找到光线与场景中某表面（Surface）的交点。鉴于本文章不是介绍如何高效地与各种几何体进行求交判断，且有很多关于光线求交的其他文章，故只给出圆盘求交[4]（Disk）和半平面（Half-Plane）求交的方法。

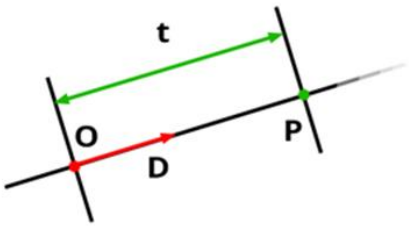


图 3 参数化表示的光线

给定一条光线 $\mathbf{p}=\mathbf{o}+\mathrm{t}\mathbf{d}$（$\vec{\mathrm{o}}$ 为起点，$\vec{\mathrm{d}}$ 为方向，如图 3），目标是找到在 $\mathrm{t}>0$ 时与起点 $\vec{\mathrm{o}}$ 最近物体的最近交点。

首先对于圆盘，其隐式方程由下式给出：

$$(\mathrm{x}-\mathrm{c}_{\mathrm{x}})^2+(\mathrm{y}-\mathrm{c}_{\mathrm{y}})^2=R^2$$

用向量代替其中分量则为：

$$\mathrm{dot}(\mathbf{p}-\mathbf{c})=R^2$$

将光线的参数表达式代入上方程，可以得到关于 t 的表达式：

$$dot(\mathbf{o}+\mathrm{t}\mathbf{d}-\mathbf{c},\mathbf{o}+\mathrm{t}\mathbf{d}-\mathbf{c})=R^2$$

化简得：

·

$$dot(\mathbf{d},\mathbf{d})t^2+2\mathrm{dot}(\mathbf{d},\mathbf{o}-\mathbf{c})t+\mathrm{dot}(\mathbf{o}-\mathbf{c},\mathbf{o}-\mathbf{c})-R^2=0$$

这是一个一元二次方程，其中判别式为：

$$\Delta=(2\mathrm{dot}(\mathbf{d},\mathbf{o}-\mathbf{c}))^2+4\mathrm{dot}(\mathbf{d},\mathbf{d})(\mathrm{dot}(\mathbf{o}-\mathbf{c},\mathbf{o}-\mathbf{c})-R^2)$$

$$\begin{cases}\Delta<0, \text{无交点}\\ \Delta=0, \text{有且仅有一个相切的交点}\\ \Delta<0, \text{有两个交点，一个进入时的交点、一个离开时的交点}\end{cases}$$

下面只需解出 t 即可：

$$t=\frac{-dot(\mathbf{d},\mathbf{o}-\mathbf{c})\pm\sqrt{dot(\mathbf{d},\mathbf{o}-\mathbf{c})^2-dot(\mathbf{d},\mathbf{d})dot(\mathbf{o}-\mathbf{c},(\mathbf{o}-\mathbf{c})-R^2)}}{dot(\mathbf{d},\mathbf{d})}$$

对于半平面的相交测试更为简单。由图 2 可以看出二维空间中的半平面就是带有法向量的直线。直线的隐式方程由下式给出：

$$Ax+By+C=0$$

类比推导求解圆盘交点的过程，设系数向量 $\mathbf{c}=(A,B)$，将光线的表达式代入可得：

$$\mathrm{dot}(\mathbf{o}+t\mathbf{d},\mathbf{c})+C=0$$

化简得：

$$\mathbf{oc}+t\mathbf{dc}+C=0$$

可解得：

$$t=\frac{-(C+dot(\mathbf{o},\mathbf{c}))}{dot(\mathbf{d},\mathbf{c})}$$

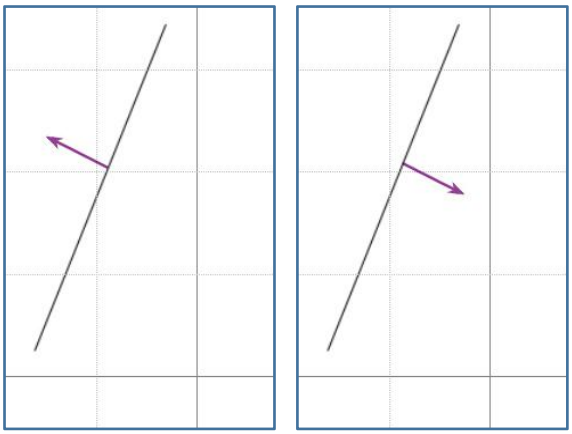


图 4 法向量决定着半平面的方向

此外还可以利用多个几何图形的交、补或差组合出更多丰富的图形，不做多余解释。

2 采样

现构建一个仅含有一个光源的场景，该光源向整个 360° 发出 $\mathrm{RGB}(255,255,255)$ 颜色的光。在第 0 节及第 1 节中讨论了如何计算每个像素的颜色值，现再次简单的总结一下。需要做的仅仅是以该点为光线起点向各个方向采样，直到遇到光源或逸出场景（逸出即未与光源相交，返回为黑色即可），最后求所有光线的平均值。最简单的是在以点 $\mathrm{p}(x,y)$ 为中心的单位圆的各个方向上均匀采样（Uniform sampling）：

```
Color uniformSampling(const Point2d & p, Scene & s, int samples)//对点 p 进行采样
{
    Color c(0, 0, 0);
    for (int n = 0; n < samples; n++)
    {
        Interaction inte;//用于记录相交时的一些信息，比如法向量、点坐标等
        Ray r = Ray(Point2d(p.x, p.y),uniformSample_in_unit_disk());//
        c += trace(r, &inte, s);//递归地追踪光线
    }
    c /= samples;
    return c;
}
```

单位圆的均匀采样，目的是在单位圆内概率相同地找出随机的一个样本。

首先是拒绝法（Rejection）。当采样的分布 $p_{complex}(x)$ 非常复杂时，可以找到另一个较简单的分布 $p_{simple}(x)$ 对其进行采样，当采样点落在 $p_{complex}(x)$ 内时接受，否则拒绝，图 5 给出了该方法的基本思路。

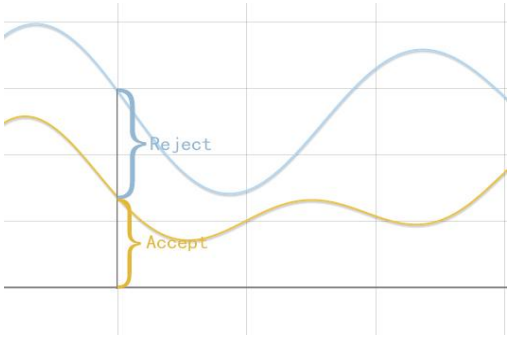


图 5 上方为 $p_{complex}(x)$，下方为 $p_{simple}(x)$

在对单位圆进行采样时，先对正方形区域 $[-1,1]^2$ 进行采样，计算该采样点 c 到圆心 $(0,0)$ 的距离，如果小于半径 1 就接受，大于半径 1 就拒绝。

伪代码如下：

```
Point result;
do
{
    result = sampling from [-1,1]^2
} while ( Length of result to (0,0) >= 1);
return result;
```

拒绝法容易编程实现，但是缺点也很明显，落在单位圆中（被接受）的概率为 $\frac{\pi}{4} \approx 0.7853$，而落在圆外（被拒绝）的概率为 $\frac{4-\pi}{4} \approx 0.2147$，因此会有接近于 21.47% 的时间被浪费在了拒绝上，导致效率降低（图 6）；此外拒绝法无法使用分层采样(Stratified Sampling)，导致收敛速度变慢因此并不常用。

·

理论上如果采用的 $p_{simple}(x)$ 越与 $p_{complex}(x)$ 接近，方差就越小，但却失去了易于采样的优点。因此还有另一种逆变换法(Inverse Transform Sampling)。

我们知道在单位圆内采样的概率密度函数为一个常数：$p(x)=\frac{1}{\pi}$。将其变换为极坐标的形式：

$p(r,\theta)=\frac{r}{\pi}$。现在计算与之对应的边缘密度函数和条件分布函数：

$$p(r)=\int_0^{2\pi} p(r,\theta(d\theta=2r$$

$$p(\theta\mid r)=\frac{p(r,\theta)}{p(r)}=\frac{1}{2\pi}$$

分别对上述方程积分求得对应的累计分布函数：

$$\mathrm{P}(r)=\int_0^r 2r'dr'=r^2$$

$$\mathrm{P}(\theta\mid r)=\int_0^\theta \frac{1}{2\pi}d\theta'=\frac{\theta}{2\pi}$$

然后求逆即可得到相应的采样点：

$$\mathrm{r}=\sqrt{\xi_1}$$

$$\theta=2\pi\xi_2$$

其中 ξ_1 与 ξ_2 为服从 $\mathrm{U}(0,1)$ 的随机变量。

得到单位圆的采样点 $\mathrm{p}_{\mathrm{sample}}(x,y)$，就可以以屏幕坐标 $\mathrm{p}(x,y)$ 为光线起点，向量 $\mathrm{p}_{\mathrm{sample}}(x,y)-\mathrm{p}(\mathrm{x},\mathrm{y})$ 作为方向向场景中递归式地发射光线了。

渲染结果正如图 7 所示。

由概率统计学，分布的平均值的标准差由下式给出：

$$\sigma_{\bar{x}}=\frac{\sigma}{\sqrt{\mathrm{samples}}}$$

其中 σ 是总体的标准差，$\mathrm{samples}$ 为样本大小。换句话说就是，随着样本大小 $\mathrm{samples}$

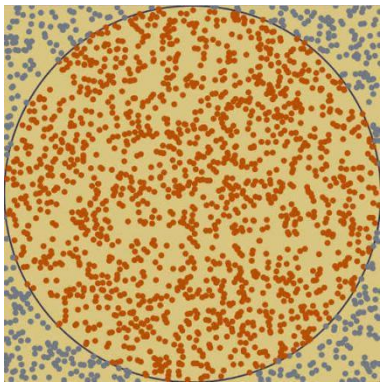


图 6 拒绝法：samples=10000 时的样点分布

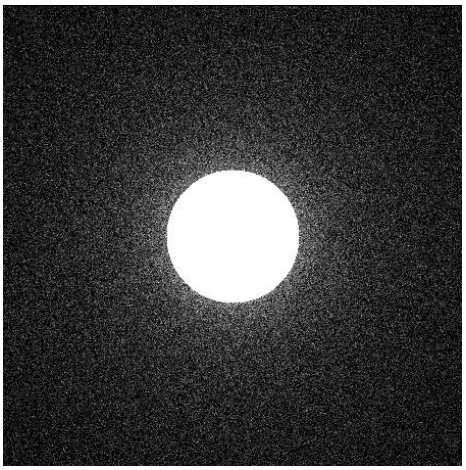


图 7 对每个点进行 8 次采样

的增加，函数将以 $O(\sqrt{samples})$ 的速率收敛。再通俗点来讲，就是每增加 4 倍的样本，方差将减少一半（图 8 展示了每增加 4 倍采样速率，噪点将减少一半）。

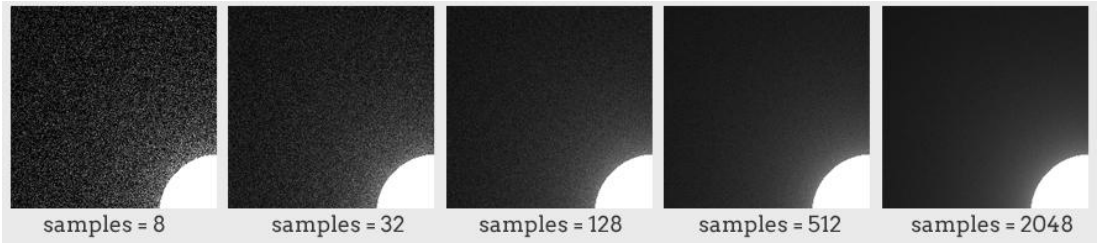


图 8 随着样本增加，噪点更少，但是渲染时间更久

可以采用分层抖动采样（Stratified Jittering Sampling）减少方差。基本思想是将总体 Ω 分成若干个次级区域 $A_1, A_2, \ldots\ldots, A_n$，每个区域称作层（Stratum）。为了从总体 Ω 采样，需要在每个不同的层 A_i 中均匀采样 $samples_i$ 个样本。对于每个不同的层，蒙特卡洛积分为

$$F_i = \frac{1}{samples_i} \sum_{j=1}^{samples_i} \frac{f(X_{i,j})}{p_i(X_{i,j})}$$

因此在实现时，可以将每点的 360° 方向划分为 $samples$ 个区域，每个区域只均匀采样 1 次，如图 9 所示。

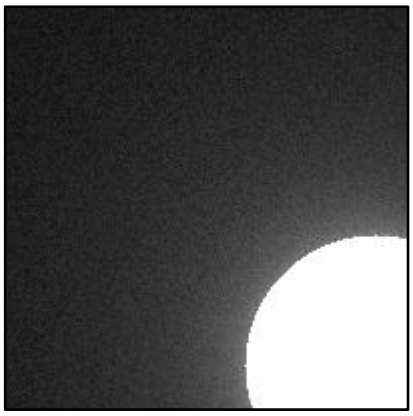


图 9 Stratified Jittered, Samples = 32

3 散射

在传统三维空间的光线传输中，使用 BSDF（Bidirectional Scattering Distribution Function）描述一对入射、出射光在表面的交互情况，BSDF 包含 BRDF 和 BTDF 两项，分别描述了反射分布和透射分布。BSDF 被定义为：

$$\frac{\text{表面反射到}\omega_i\text{方向的微分辐射率}}{\text{表面上来自入射光方向}\omega_o\text{的微分辐照度}}$$

即：

$$f(\omega_o, \omega_i) = \frac{dL(\omega_o)}{dE(\omega_i)}$$

提供入射光，并对 BSDF 采样即可获得与之对应的出射光。

由于是在二维空间中模拟光，无法直接观察到表面本身与光线交互后产生的效果，比如光泽、粗糙度，故假设场景中所有的物体都是绝对光滑且分布均匀没有杂质的，此时 BSDF 是一个 Delta 函数，仅在某个方向能取到值，这样能直观的观察到反射与折射在二维空间中

·

的效果，也降低了实现难度。

当光线与表面发生交互时，该束光携带着不同波长的光子会发生如下三种情况：穿透、吸收、散射。当物体为导体且绝对光滑时光子均被反射或吸收；当物体为电介质时且分布均匀且无杂质时，光子均被透射到表面的另一面或被吸收。其中被吸收的不同波长的光子数量不同在宏观表现上就体现在了物体表面的颜色。图 10 说明了光子与物体交互时将会发生的三种情况。

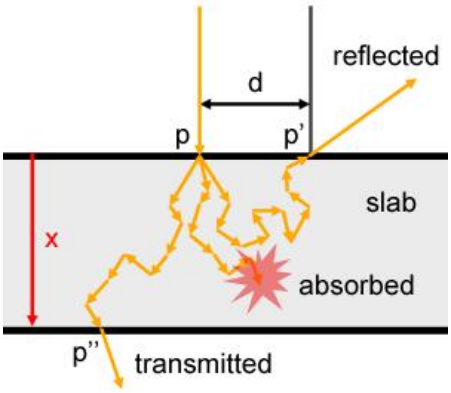


图 10 光子的穿透、吸收以及反射

对于反射向量的计算，其反射光线的方向遵循着：入射角 = 出射角。假设 $\mathbf{n}$ 与 $\mathbf{d}$ 均已标准化。

如图 11 所示，反射向量 $\mathbf{r}$ 满足：

$$\mathbf{r} = 2\mathbf{e} + \mathbf{d}$$

现要求解 $\mathbf{e}$，首先要计算出 $\mathbf{p}$ 向量：

$$\mathbf{p} = dot(\mathbf{n}, \mathbf{d})\mathbf{n}$$

因此

$$\mathbf{e} = -\mathbf{d} + \mathbf{p} = dot(\mathbf{n}, \mathbf{d})\mathbf{n} - \mathbf{d}$$

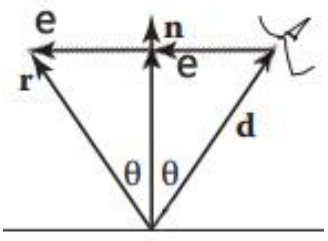


图 11 反射向量

代入 $\mathbf{r} = 2\mathbf{e} - \mathbf{d}$ 可得到反射向量 $\mathbf{r}$ 的计算公式：

$$\mathbf{r} = 2(dot(\mathbf{d}, \mathbf{n}))\mathbf{n} - \mathbf{d}$$

在实现时，由于程序运行时的入射向量与此处分析时的入射向量相反，故需要添加一个负号（计算折射向量时同理）。代码如下：

```
Vector reflect=  d - 2 * Dot(n, d) * n;
```

计算出反射向量后，就可以递归式地进行追踪光线了。要注意的是应该将表面的反射率考虑在内。现实中反射率应当是定义在波长上的一个连续函数，此处仅取 RGB 三种波长，意为分别对红、绿、蓝三种颜色的反射率。反射率范围应处于 $[0,1]$，否则能量不守恒。

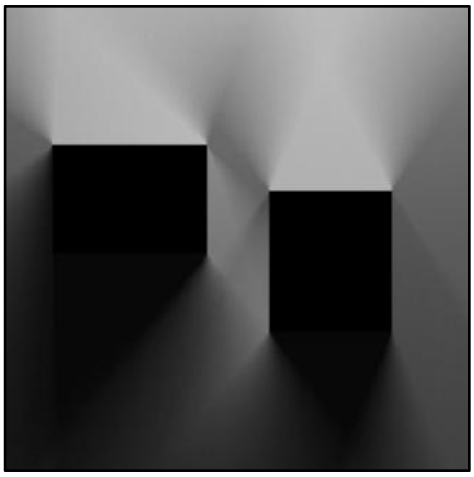


图 12 反射

渲染结果如图 12 所示。此外由图 12 可以看出产生了软阴影，这是由于几何遮挡关、系并且光源不是理想点光源而形成的。

除了前文所提到的反射，还有一种物质可以透射光线，这种物质又称为电介质（Dielectrics）。当光线从折射系数（IOR）（此处为区别于 refractivity，将 IOR 称为折射系数）为 η_L 的介质向折射系数为 η_T 的介质传播时，会发生透射与弯曲，其中 η_T 相对于 η_L 越大，弯曲程度越大。

·

参考图 13，对折射向量的推导过程如下：

斯涅耳定律（Snell’s Law）给出了入射角 θ 与出射角 ϕ 的关系：

$$\eta_L \sin\theta_L = \eta_T \sin\theta_T$$

假设 $\mathbf{N}$ 与 $\mathbf{L}$ 均已标准化，则折射向量 $\mathbf{T}$ 可以表示为为平行于 $\mathbf{N}$ 的 $-\mathbf{N}\sin\theta_T$ 和垂直于 $\mathbf{N}$ 的 $-\mathbf{G}\sin\theta_T$ 的线性组合。由图 13，可得到 $\mathbf{G}$：

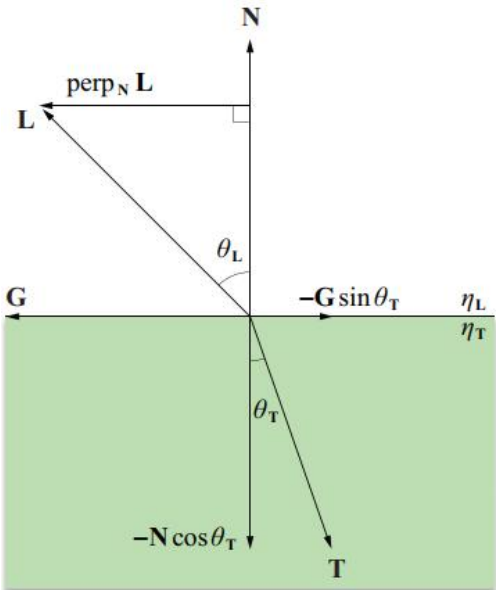


图 13 折射向量示意图

$$\mathbf{G} = \frac{\text{perp}_{\mathbf{N}}\mathbf{L}}{\sin\theta_L} = \frac{\mathbf{L} - \text{dot}(\mathbf{N},\mathbf{L})\mathbf{N}}{\sin\theta_L}$$

反射向量 $\mathbf{T}$ 可以表示为两个分量的和，即：

$$\mathbf{T} = -\mathbf{N}\cos\theta_T - \frac{\mathbf{L} - dot(\mathbf{N},\mathbf{L})\mathbf{N}}{\sin\theta_L}\sin\theta_T$$

利用三角等式以及斯涅耳定律做代换化简可得：

$$\mathbf{T} = \left(\frac{\eta_L}{\eta_T}\text{dot}(\mathbf{N},\mathbf{L}) - \sqrt{1-\left(\frac{\eta_L}{\eta_T}\right)^2[1-dot(\mathbf{N},\mathbf{L})^2]}\right)\mathbf{N} - \frac{\eta_L}{\eta_T}\mathbf{L}$$

当且仅当光从光密介质向光疏介质发射时（即 $\eta_T > \eta_L$），会发生全反射（Total Reflection），即光线按照反射公式在内部再次被反射回内部，此时上述计算折射向量的公式中 $\sqrt{1-\left(\frac{\eta_L}{\eta_T}\right)^2[1-dot(\mathbf{N},\mathbf{L})^2]}$ 项平方根内的数值为负数。在编写代码是要注意处理全反射：

```
{
    ...
    if (underSquareRoot < 0.0f)
        Deal with total reflection;
    ...
}
```

由图 14 可以看到渲染出了非常漂亮的焦散(Caustic)，这在三维空间中使用传统路径追踪几乎无法实现。事实上传统的路径追踪是可以渲染出焦散的，通过采样 BSDF，但是正如第 0 节中介绍的，三维空间中某点的辐射率来自于整个上半球空间的积分，因此采样 BSDF 时只有少部分路径是与焦散的路径相吻合，换句话说焦散在上半球空间上是有一个形状的，但是与 BSDF 的形状相差很远，因此渲染效率及其低下，甚至需要上万条采样才可以渲染出

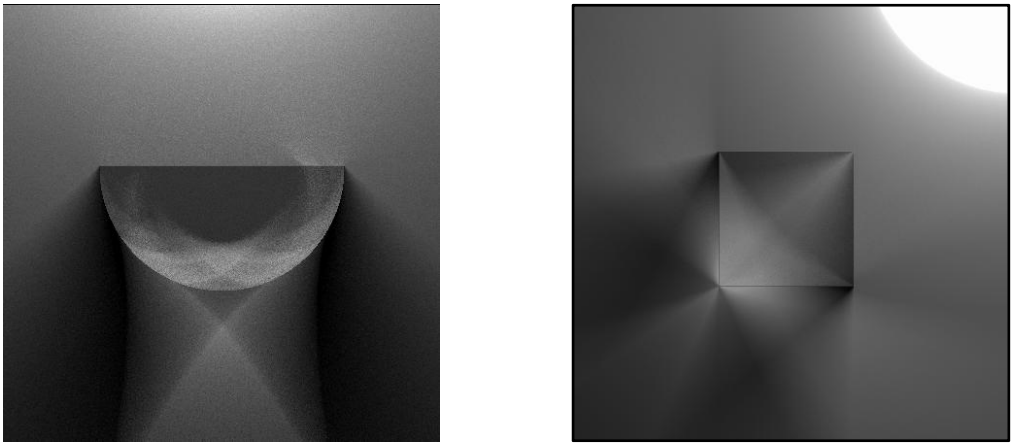


图 14 折射

焦散。而二维空间中，正如第 1 节介绍的二维渲染方程，仅需要从 360° 各个方向进行采样，相比于上半球空间，方差更小，因此在二维空间中的全局光照计算可以轻松模拟出焦散效果。

与上述同样的原因，在三维空间中难以渲染的色散(Dispersion)，在二维空间中也较为轻松，但本文并不予实现色散效果。

现实中的，光线从介质 1 进入到介质 2 时，在两者之间会同时发生反射与折射，随着视角、折射系数不同，对应的反射率和折射率均不同。菲涅耳方程（Fresnel Equations）则给出了反射率关于折射系数、入射角、出射角的关系式，折射率可以通过

$refractivity = 1 - reflectivity$ 获得。

菲涅耳方程与与光波的两个互相正交的偏振光有关，因此菲涅耳方程有两组，分别对应 s 方向偏振和 p 方向偏振。

$$R_{\mathrm{s}} = \left| \frac{n_1 \cos\theta_{\mathrm{i}} - n_2 \cos\theta_{\mathrm{t}}}{n_1 \cos\theta_{\mathrm{i}} + n_2 \cos\theta_{\mathrm{t}}} \right|^2 = \left| \frac{n_1 \cos\theta_{\mathrm{i}} - n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_{\mathrm{i}} \right)^2}}{n_1 \cos\theta_{\mathrm{i}} + n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_{\mathrm{i}} \right)^2}} \right|^2,$$

$$R_{\mathrm{p}} = \left| \frac{n_1 \cos\theta_{\mathrm{t}} - n_2 \cos\theta_{\mathrm{i}}}{n_1 \cos\theta_{\mathrm{t}} + n_2 \cos\theta_{\mathrm{i}}} \right|^2 = \left| \frac{n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_{\mathrm{i}} \right)^2} - n_2 \cos\theta_{\mathrm{i}}}{n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_{\mathrm{i}} \right)^2} + n_2 \cos\theta_{\mathrm{i}}} \right|^2.$$

该公式计算复杂，且在计算机图形学中通常不考虑偏振光的影响，Christophe Schlick[10]给出了对该公式的近似：

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

$$R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2} \right)^2$$

其中 R_0 是光线垂直于表面时的反射率。

图 15 展示了加入了菲涅尔效果后各 IOR 的对比。

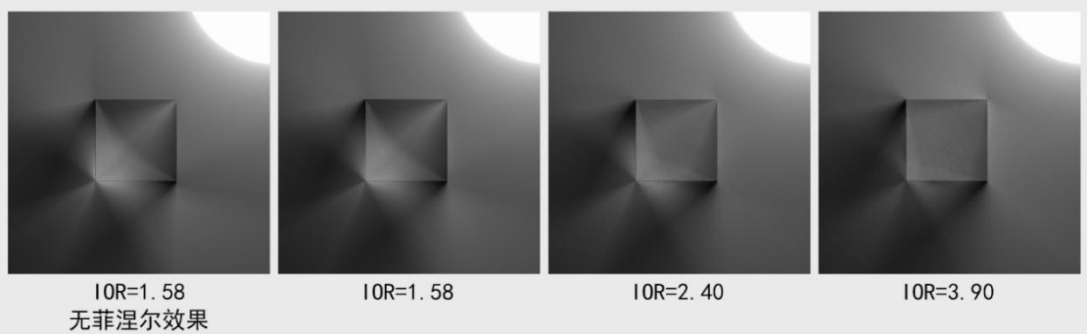


图 15 注意观察靠近光源的物体边缘，反射率与 IOR 成正比

3 体散射

在第 3 节中提到了一束光的光子在表面交互时的三种过程。现实中，光线穿过任何物质都会发生上述三种过程，但由于真空不含任何粒子，所以真空不会发生上述情况，前文中的反射和折射均假设在真空中发生。例如烟（图 16）、雾、云。

图 16 黑色的烟，其中的粒子吸收了大部分的光子，所以颜色更深

当介质中的粒子主要是吸收光子的时候，其外表颜色将表现得更深一点。使用吸收系数σ_a来表示介质的对光子吸收的能力。在现实中，吸收系数是会随着光线位置、方向的不同而不同，因此它本质是一个分布函数。此处假设介质中的粒子各向同性，即σ_a为一个常数。假设一束光线携带着辐射率$\mathrm{L_i}$从点$\mathbf{p}$向方向$\boldsymbol{\omega}$发射并在介质中传播了距离d，那么经过介质后剩余的辐射率为：

$$\mathrm{L_o} = \mathrm{e}^{\int_0^d \sigma_a(\mathbf{p}+t\boldsymbol{\omega},\boldsymbol{\omega})dt}$$

光子不仅仅会被粒子吸收，撞上粒子时也有概率会被散射。与吸收类似，光线中的光子被散射后光线携带的辐射率也会降低，不同的是光子被散射到了其他方向，而非被吸收。和吸收系数一样，散射系数σ_{s}也是一个关于位置和方向的分布函数，此处假设为常数。

吸收系数和散射系数统称为物质对光的衰减系数（Attenuation Coefficient）σ_{t}：

$$\sigma_t = \sigma_{\mathrm{a}} + \sigma_s$$

其中σ_{a}、σ_{s}、σ_{t}的单位均为(m^{-1})，即这三个量的取值范围均为$[0,\infty]$

下式给出了光线经过各向异性介质从点$\mathbf{p}$到点$\mathbf{p}'$，透射率T与衰减系数、光程d的关系：

·

$$T(\mathbf{p},\mathbf{p}') = e^{-\int_0^d \sigma_t(\mathbf{p}+t\boldsymbol{\omega},\boldsymbol{\omega})dt}$$

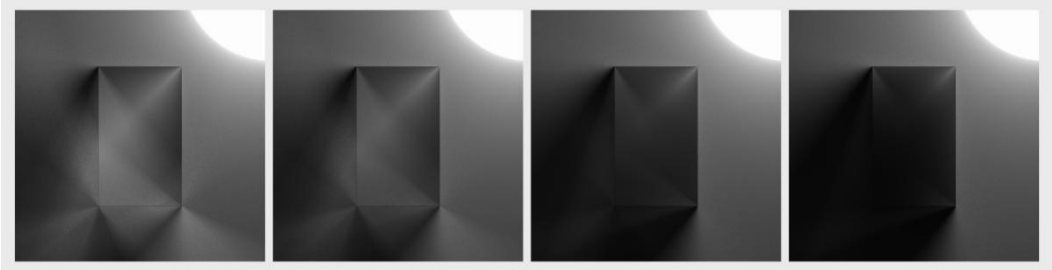
将透射率与入射光线携带的辐射率相乘，可得到经过介质后的出射辐射率。而两点间的光学厚度（Optical Thickness）由下式给出（即上式中积分里的内容）：

$$\tau(\mathbf{p},\mathbf{p}') = \int_0^d \sigma_t(\mathbf{p}+t\boldsymbol{\omega},\boldsymbol{\omega})dt$$

比尔-朗伯定律（Beer-Lambert law）则给出了在各向同性介质中的透射率：

$$T(\mathbf{p},\mathbf{p}') = e^{-\sigma_t d}$$

对电介质应用比尔-朗伯定律的渲染结果如图 17 所示。



$\sigma_t = 0.0$　　$\sigma_t = 2.0$　　$\sigma_t = 8.0$　　$\sigma_t = 16.0$

图 17 比尔-朗伯定律

此外 $\frac{\sigma_s}{\sigma_t}$ 被称为反射率（Albedo）。反射率取值范围在 $[0,1]$，描述了反射的概率大小；$\frac{1}{\sigma_t}$ 被称为平均自由程（Mean Free Path），描述了光线在介质中与粒子发生碰撞前的平均路程。

光线与表面交互时，出射光通过采样 BSDF 获得。而与介质交互时，通过采样 Phase Function[8]获得。Phase Function 给出了给定波长的光以角度 θ 为自变量的光照强度分布。与 BSDF 一样，Phase Function 不止有一种。

最常用的 Phase Function 是由 Henyey 和 Greenstein 提出，并以他们的名字为名的 Henyey-Greenstein Phase Function[9]：

$$p_{H-G}(\theta) = \frac{1}{4\pi}\frac{1-g^2}{[1+g^2-2g\cos(\theta)]^{3/2}}$$

其中 g 被称为非对称参数（Asymmetry Parameter），$g \in [-1,1]$。当 $g<0$ 时，对应于后向散射（Backward Scattering）；当 $g>0$ 时，对应于正向散射（Forward Scattering）。这两种情况（图 18）均为各向异性散射。当 $g=0$ 时，为各向同性，向各个方向散射的概率相

·

等。

对该函数采样，使用逆变换法。首先对该函数积分求得 CDF：

$$P(\cos(\theta)) = \frac{1}{2}\int_{-1}^{\cos\theta} \frac{1-g^2}{[1+g^2-2g\cos(\theta)]^{3/2}}$$

$$= \frac{1-g^2}{2g}[(1+g^2-2g\cos(\theta))^{-\frac{1}{2}} - (1+g)^{-1}]$$

对其求逆，可得到 $\cos(\theta)$：

$$\cos(\theta) = \frac{1}{2g}[1+g^2-(\frac{1-g^2}{1-g+2g\xi})]$$

其中 ξ 为服从 $U(0,1)$ 的随机变量。

得到了 $\cos(\theta)$ 通过三角关系求出 $\sin(\theta)$，这样就得到了光线的新的方向。

利用上述方法可以渲染出图 19 中体积（Volume）的效果.

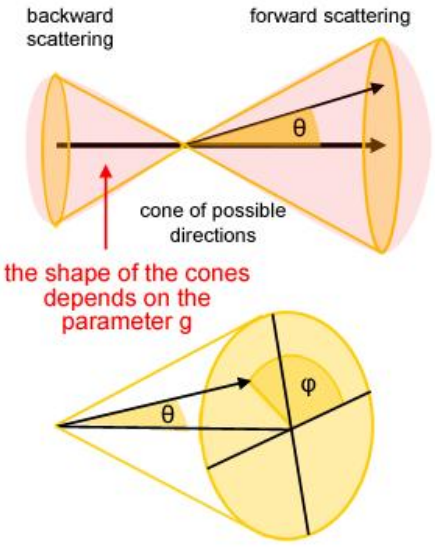


图 18 前向散射与后向散射

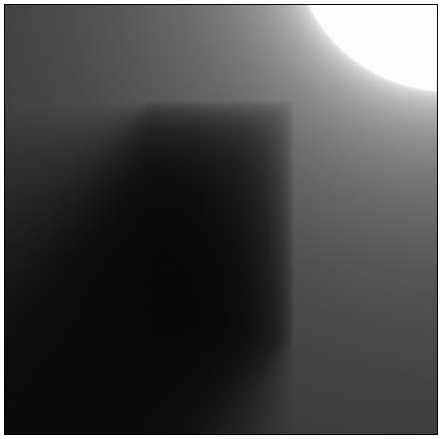


图 19 体渲染

参考文献

[1] Palmer, James M. "The SI system and SI units for Radiometry and photometry". Archived from the original on August 2, 2012.

[2] 盛骤、谢式千、潘承毅，《概率论与数理统计》（第四版）第五章，高等教育出版社，2008。

[3] Matt Pharr, Wenzel Jakob, Greg Humphreys, Physically Based Rendering: From Theory To Implementation, 3rd Edition, 2016.

[4] Steve Marschner, Peter Shirley, Fundamentals of Computer Graphics, Fourth Edition-A K Peters, Limited, Taylor & Francis Group [distributor_AK Peters (2016).

[5] Botev, Z.; Ridder, A. (2017). "Variance Reduction". Wiley StatsRef: Statistics Reference Online: 1--6. doi:10.1002/9781118445112.stat07975.

[6] Wikipedia, "Inverse transform sampling",website: https://en.wikipedia.org/wiki/Inverse_transform_sampling

[7] Eric Lengyel, Mathematics for 3D Game Programming and Computer Graphics, 2002.

[8] Liou, K. N., 2002. “An Introduction to Atmospheric Radiation”. Academic Press c/o Elsevier Science, San Diego, CA.

[9] Henyey, L.G., and Greenstein, J.L (1941), Ap.J., 93, 70.

[10] Schlick, Christophe. "An Inexpensive BRDF Model for Physically-based Rendering." Computer graphics forum. Vol. 13. No. 3. Blackwell Science Ltd, 1994.