**The Rendering Equation**

Author: James T. Kajiya

ACM, 1986

Translator:练孙鸿

1. **摘要**

我们提出了一个积分方程来用来概括和泛化已知的渲染算法。在讨论蒙特卡洛方法(Monte-Carlo solution)的同时，我们还提出了一种新的方差减少(variance reduction)的方法：分层采样(Hierarchical Sampling)。然后我们给出了一些精巧的说明来说明这种新的采样方法能使得蒙特卡洛方法变得更有效率。本文提出的渲染算法框架包含了很大范围的光学现象，且用此算法可以有效地模拟这些现象。

1. **渲染方程**

我们提出的这一个方法囊括了非常多的渲染算法，并且**把它们都统一到了同一个背景下，并把它们看作是同一个方程的不同精度的近似解**。但把众多渲染算法统一到一个框架在却无须惊讶，因为所有的渲染方法都旨在对同样的物理现象进行建模：也就是光在各种物体表面上的散射。

我们也需要指出渲染方程(rendering equation)背后的思想不是完全创新的。在辐射热传递的专著里面[Siegel and Howell 1981]，渲染方程所模拟的现象已经被研究得比较透彻了。但是我们提出的这种形式的方程却是非常适合计算机图形学的，这种形式的方程以前从未出现过。

渲染方程(1)是：

其中：

与光的强度有关，从点传递到

是几何项(“geometry” term)

与自发光强度有关，从点

与光的散射强度有关。从入射到一小块表面区域，然后散射到出的光强度。

这个渲染方程在很大程度上本着光能传递方程的精神，平衡了从表面上一个点流向另一个点的能量。这个方程表明了，从物体表面上一个点到空间中另外一个点的光照的传输强度，就等于自发光加上来自于其他方向的光散射之后的出射光。而公式(1)与光能传递方程也是不一样的，(1)没有对物体表面的反射特性做出任何假设。这个积分的定义域是全场景的表面，因为会分布在全场景的物体表面上。当然我们还会有一个全局的背景（如天空球、天空盒），用来包含整个场景。

作为对电磁学里面的麦克斯韦方程组(Maxwell’s Equation)的一种近似，(1)并没有打算对所有的光学现象进行建模。(1)是一种**几何光学的近似**，它只对一定时间内的平均传输能量进行建模，所以衍射(diffraction)不在考虑范围之内。我们也假设表面两面的介质具有均匀的折射率(homogeneous refractive index)，且他们自身不参与散射。如果介质要参与光能传输，我们就需要引入积分-微分方程(integro-differential equation)了。参与传输的介质的模拟和衍射的相位计算需要用路径积分(path integral)的方法来解决。最后，**方程(1)与光的波长和偏振无关**。

1. **关于传输的物理量**

我们现在来讨论下关于(1)中的物理量。首先，这个方程用简化的模型描述了光子传播的强度。测量了从到传播的辐射能，我们给命名为从到的无遮挡两点传输强度，或者简化地描述：传输强度。传输强度是单位时间内、单位光线发射源处的面积、单位目标出的面积的辐射能。

所以I的单位是。

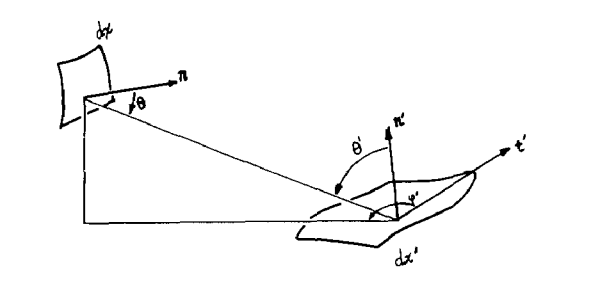
是几何项，这一项表示了表面点是否有被遮挡。如果光路有遮挡，此项为0；如果无遮挡，则此项为。

表示的是表面点出发出的光（自发光）：

的单位是。

最后来看散射项（译者注：也就是BxDF），它表示了从发射，入射到，并散射到处的光强度。这是一个无量纲的项：

一般情况下，辐射强度（radiometric intensity）定义为单位投影面积、单位立体角(solid angle)、单位时间内的能量(5)：



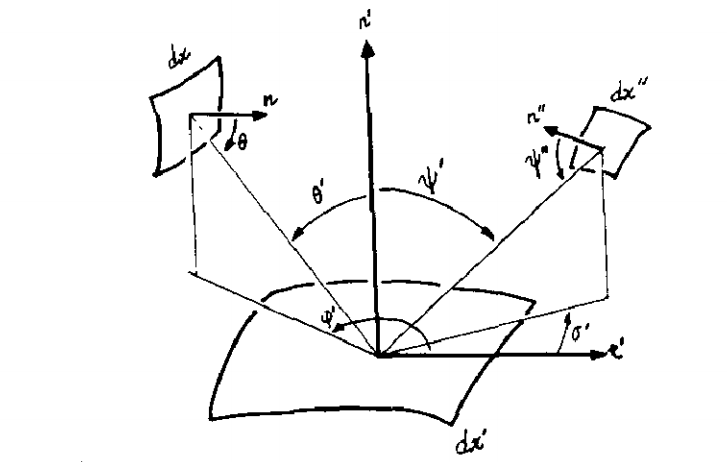
从图中我们可以推导出：

其中分别是表面单元的法线，是表面单元的切线，。被面积单元正对着的立体角，等于的投影面积除以圆的半径：

所以：

那么”辐射强度”可以写成：

最后我们把传输的反射强度(transport reflectance)和普通的辐射度学的双向全反射函数(total bidirectional reflectance function)联系在一起：

**

从图中可以得到：

最后我们得到了基于辐射度学反射强度的双向全反射函数：

1. **近似的解决方案**

在这一节，我们会分析一下渲染方程的近似解决方案，各种解决方案都可以被统一在渲染方程这个框架下面。有很多其他的渲染方案还没有囊括在这项工作中，但我们会分析好几种渲染方案。

在此之前我们先引入**Neumann级数**(Neumann Series)。在物理数学方法里面，有一种求解积分方程的办法，就是用Neumann级数对符合某种形式的线性算子(linear operator)进行展开。众所周知，积分(integral)是一个线性算子，它满足：

那么我们可以把渲染方程这个积分方程写成一个抽象的算子形式：

其中就是一个线性算子，其具体形式可以参考渲染方程本身。然后我们再进行变式，得到：

其中1是单位元算子。假设是赋范向量空间(normed vector space)的有界线性算子，如果Neumann级数可以收敛于算子范数(operator norm)，那么可逆且他的逆是：

其中是向量空间中的单位元算子(identity operator)。那么把Neumann级数的展开用在渲染方程上，我们有(2)：

其中只需要算子的半径小于1，这个无穷级数就会收敛。而且这个Neumann级数展开也有物理意义：**级数展开式的每一项对应着光线的第次散射**。接下来介绍几种渲染方程近似解决方案。

* **“Utah”近似**

由于没有很好的名字，我们把“那个“经典的渲染算法叫做Utah近似(Utah Approximation)。按照译者的理解，Utah Approximation就是只处理直接光照的经典光栅化、局部光照(local lighting)的渲染。这种渲染近似就只有两项：

Utah Approximation只考虑了光线的第一次散射，而且它的几何项的计算比较困难。之所以Utan Approximation的第二项的算子是直接作用在上而不是，是因为局部光照算法在初次着色的时候直接忽略掉可见性(visibility)，也就是直接着色**忽略掉阴影**。而意思是这种算法只支持精确光源（如平行光、点光源、聚光灯），这样子在进行光照着色的时候就不需要使用积分，而是单纯地求和（因为使用精确光源的时候，入射到表面某一点的光线只有一根）。这么多年来，有很多对这种渲染方法的拓展技术，例如Shadow Mapping，Shadow Volume等阴影技术、IBL等拓展的光照技术。

* **光线跟踪(ray tracing)**

[Whitted80]提出了一个基于光线跟踪的著名的近似渲染算法：

在这篇突破性的论文里面，是一个简化的表面散射模型。但这个渲染算法依旧还局限着精确光源的使用，所以多次散射的项还是在使用，而第一项环境光项用的是。

* **分布式光线跟踪(distributed ray tracing)**

[Cook et al 1984]中提出了分布式光线跟踪，这一种近似方法提出了更加精确的散射模型，且要求使用积分来求解算子。这项新的工作让我们可以开始使用Monte-Carlo方法来求解渲染方程的积分项。但是这篇论文依旧没有处理环境光照。

* **光能传递算法(Radiosity)**

[Goral et al 1984, Cohen and Greenburg 1985, Nishita and Nakamae 1985]把光能传递算法引入了计算机图形学。这在80年代来说是一个全新的渲染算法，它能很好地处理漫反射表面的能量平衡方程(energy balance equation)。漫反射表面指的是不依赖双向反射函数(bidirectional reflectance function)的角度变量，也就是：

表面微元的光能传递(radiosity) 就是处的可视(visible)正半球面的能量通量(energy flux)。他是单位面积上单位时间的能量，单位是。它的定义是(15)：

其中是半球面。所以为了计算半球的物理量，我们要在场景所有的表面上积分。结合(1)和(15)我们有：

如果的有遮挡，则其对出射率的贡献为0，不然的话(16)：

其中的半球面出射率为。

其中反射项也是类似的(17)：

其中是半球面上单位时间单位面积的入射能量。我们把(17)和(18)放进(16)里面，渲染方程便变成了(19)：

这个就是[Goral et. al. 1984]里面的方程(4)。

1. **分层采样(Hierarchical sampling)**

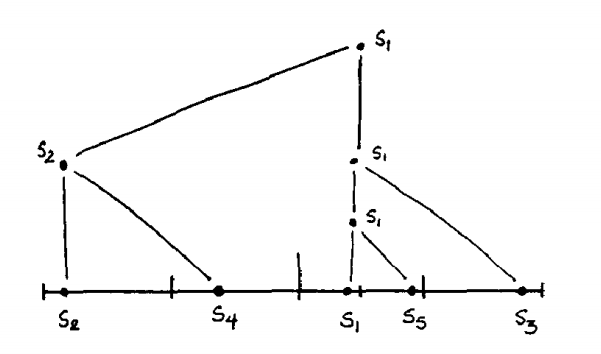
我们现在提出一些新的、用于求解渲染方程的方差减少(variance reduction)技术。同时我们也要指出，这里提及的方差减少技术可以应用在更加广阔的场景里面。总的来说，这些方差减少技术在各种蒙特卡洛积分(Monte-Carlo Integration)问题里面都有着很重要的应用。很多蒙特卡洛积分的被积函数(integrand)不一定有闭合形式，或者其形式非常复杂，导致积分的计算非常困难。在这种情况下，只使用已知的方法带来的开销已经变得不可忽略了。所以这里我们提出五种方法，让蒙特卡洛积分里面有限数量的采样发挥他们最大的作用。下面介绍的方法的思想都来自于**分层采样(stratified sampling)**。

* **序列均匀采样(Sequential uniform sampling)**

第一种采样技术起源于一种常见的序列采样策略。一般情况下，被积函数的采样被不断地生成，并加入到最终的结果中，直到积分的采样方差降到一定的阈值之下，我们才会停止继续采样。这种策略在[Lee, Redner and Uselton 1985]里面展示出了一定的优势，在论文里面，他们只对感兴趣的部分投放大部分的采样，而不感兴趣的部分则只投放少量的采样，做到有一定的针对性。

但不幸的是，这种序列采样方法跟分层采样不太兼容。在分层采样技术里面，感兴趣的区域会被分成很多个子单元(subcells)。Lee等人用了每像素8个子单元的分割方法，然后分别在每个子单元里面都进行一次采样。理想的情况下，因为子单元都是被均匀分割的，所以只要每个子单元里面随机采样一次，就可以得到更好的收敛结果——这也是所谓的**抖动采样方法(jitter sampling method)**，因为我们也可以把子单元的中心点看作是一系列构成晶格(lattice)的点。那么由于序列采样方法无法事先知道需要多少个采样，从而导致无法得到要均匀分割出的子单元数量——这也就是序列采样和抖动采样不兼容的问题根源所在。

序列均匀采样(Sequential uniform sampling)却把序列采样和抖动采样结合在了一起。这种技术通过维护一棵拥有不同大小的单元的树来实现。每次投放一个采样，我们就选一个单元，并且把它分成更小的子单元。之前已经投放过的采样必须也要在新分割的子单元内，然后新投放的采样就可以放在旧采样所在子单元隔壁的子单元内。



上图是投放的五个采样的序列均匀采样树。如果要添加更多的采样，我们只需要遍历这一棵树，直到找到叶结点。然后我们对把这个叶结点一分为二，生成两个新的叶结点，然后把新的采样放到空的新叶结点里面。

但是为了保证更加均匀的细分，我们可以考虑使用宽度优先搜索(BFS, Breath First Search)。先把每一层的节点都分裂完并装满了采样再分裂出更深一层的节点。但是普通的BFS会导致不随机的采样分布，所以可以考虑用随机顺率的BFS来选择分裂的叶结点。

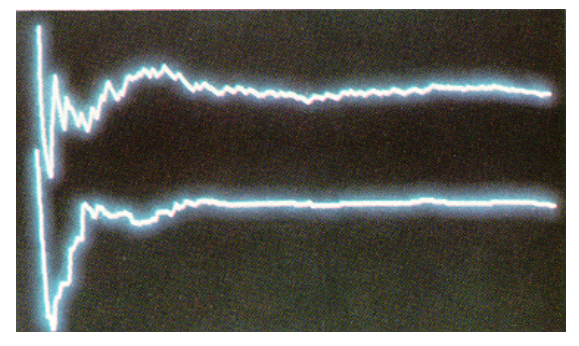
* **高维情形(multi-dimensional case)**

上面的算法可以很轻易地推广到高维空间，只需要使用一些高维空间适用的数据结构：k-d tree。在这个数据结构里面，空间区域被垂直于坐标轴的超平面(hyperplane)不断地分割。那么用kd树对路径空间的超平面分割的泛化应该也是很直观的。

* **分层积分(Hierarchical integration)**

第三种方法充分利用了每个采样就代表了一个子单元这么一个事实。在这种方法里面，我们可能会利用子单元的体积及其里面所投放采样的值来计算黎曼和(Riemann Sum)。[Yakowitz et. al. 1978]提出了一种这种方法的变体，即把采样本身当作边界且没有使用分层(stratification)。Yakowitz的这种方法在一维的情形能达到级别的方差，二维的情形达到级别的方差，这些都远比最基础的蒙特卡洛方法的要好。对于我们这个方法的分析还需要进一步的讨论，预计会有相关的论文发代表。但是由于我们采样的分层，初步估计我们的方法在积分里面的表现会更加的优秀。

每当一个子节点的单元被分割，**他对最终积分的结果的贡献就会减半**。新的被积函数采样会被乘以空格子的体积。在分裂节点和采样都完成以后，我们就遍历这棵树，生成遍历路径，使得每个节点的积分结果等于它的子节点的积分结果之和。因为我们用一棵树保存了所有内部节点的积分，所以我们才可以用这些采样的密度去自动缩放他们对积分结果的贡献，使得最终的积分结果的估计保持恒定。



图：上面是Naïve Monte Carlo积分，下面是Hierarchical Monte Carlo积分

* **自适应分层积分(Adaptive Hierarchical integration)**

第四种方法除了定义域上的采样的均匀性外，还考虑了其他的一些标准。在这种变体里面，我们把采样都集中投放在感兴趣的区域上，然后在那些值几乎是常数的区域投放稀疏的采样。

（译者注：译者认为那就是要在诸如区域光边缘之类的这种radiance剧烈变化的高频区域要多一些采样）。

我们正在寻求一种标准，用以决定哪些树的节点是我们感兴趣的部分，以及这个标准如何融入到我们的算法里面。这很容易让人想到序列均匀采样(sequential uniform sampling)的子节点选择规则。

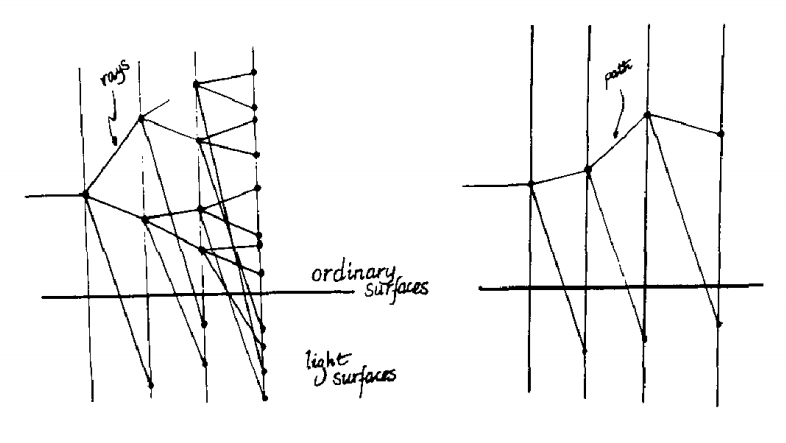
（译者注：其实没太看懂是在搞什么，这一小个方法文中也说了不太理想，所以先跳过）

* **非均匀采样：类重要性采样(Non-uniform sampling: Importance sampling analog)**

最后一种方法考虑了**重要性采样(importance sampling)**。我们之前的方法都是把子节点的格子对半分，分割得很均匀。但是我们其实可以沿着可以代表着某种概率密度函数(pdf, probability density function)的中位数(median)的超平面进行分割。超平面的选择可以在第二种方法的k-d树的超平面里面选取。把概率密度表示为概率分布函数(Integrated distribution function)让我们更加容易去求得这个中位数超平面：要找到概率密度的中位数，我们只需要找到概率分布函数处的点。

重要性采样是一种很重要的方差减少技术，对于解渲染方程来讲非常的关键。

上面讲的几种方法都可以用在各种渲染算法上面。例如，序列均匀采样技术就可以用来采样景深模糊的光圈；自适应分层积分可以用来多种采样一个像素；重要性采样可以用来选择下一条要发射的光线。把**路径跟踪**与传统的光线跟踪做比较是一件很有趣的事情，其实传统的光线跟踪也可以很容易地转换我们的路径跟踪解决方案。我们并没有去求解每个表面微元上面发出的光线，我们的解决方案只跟踪其中一条光线。而且我们的光线只射向已知的光源。



（**译者注：左边是传统的光线跟踪，从光源发射光线时的求解复杂度；而右边则是路径跟踪，从观察者出发**）。有一个非常重要的现象需要被指出来的，由于物体表面的被动性(passivity)，我们知道第一代(the first generation)的光线是最重要的，因为光的直接照射一般比其他表面散射过来的光线对最终亮度积分的贡献要大。第二代的光线对方差的贡献已经严重减少了。**但是传统的光线跟踪却把绝大部分对方差贡献极低的光线都算上了**，因为它发射了过多散射太多次数的光线了，而我们的方法却没有这样的缺点。

而且上面的图示也给出了分布式光线跟踪的一个替代方案。与其射出一棵需要分裂的采样树，不如射出选定的概率密度的射线。对于有反射和折射的场景来说，path tracing可以省掉非常多的计算量。但是要注意的是，**我们必须仔细计算反射、折射、阴影对最终积分结果的贡献比例**。

**引用**

