

光传输与蒙特卡洛方法基本介绍

Dezeming Family

2022 年 9 月 2 日

DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [<https://dezeming.top/>] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

本文是对 [1] 论文的详细解读，该论文可以说是渲染必读论文，但有些符号表示和描述可能对初学者并不友好，由于里面介绍的几种重要技术，例如双向路径追踪、多重重要性采样以及 Metropolis 方法都是非常重要的，因此我打算写一下本论文的解读，作为构建“高端图形渲染学习体系”的一个重要组成部分。

由于论文 [1] 篇幅过长，为了减少 Latex 编译的时间以及更好把控不同部分的内容，我将整个论文划分为了多本小册子来进行讲解。本文由于内容并不复杂（有些复杂的但非必要掌握的额外知识，比如一些测度论原理的证明，我们并不需要掌握，只需要知道基本概念即可），所以解释的内容并不会很多，有些没有太大必要了解的内容我也进行了删减，并不会影响阅读体验。

本文的预备知识：**蒙特卡洛方法、蒙特卡洛光线追踪（可以看 Peter Shirley 的光线追踪三本小书）、BSDF 模型、路径追踪。**

目录

一 论文的基本介绍	1
1.1 双向传输理论	1
1.2 更好的蒙特卡洛技术	1
1.3 鲁棒的光传输算法	1
二 光传输算法	2
2.1 蒙特卡洛方法与确定性方法的对比	2
2.2 视角相关与视角无关算法	3
2.3 无偏与一致蒙特卡洛算法	3
三 光学模型与它们对图形学的影响	4
3.1 几何光学	4
3.2 波动光学	4
3.3 量子光学	5
四 蒙特卡洛方法	5
4.1 求积分的法则的收敛速度上限	5
4.2 概率理论基础	6
4.3 蒙特卡洛方法	7
4.4 采样随机变量	8
4.5 估计器的性能	8
五 减少方差的方法	9
5.1 解析积分 (Analytic integration)	9
5.1.1 使用期望值	9
5.1.2 重要性采样	10
5.1.3 控制变量法	10
5.2 均匀样本放置 (Uniform sample placement)	10
5.2.1 Stratified sampling	10
5.2.2 Latin hypercube sampling	11
5.2.3 Orthogonal array sampling	11
5.2.4 Quasi-Monte Carlo methods	11
5.3 自适应样本放置 (Adaptive sample placement)	12
5.3.1 自适应采样	12
5.3.2 Russian roulette	12
5.3.3 splitting	12
5.4 相关的估计器 (Correlated estimators)	13
5.4.1 Antithetic variates	13
5.4.2 Regression methods	13
六 本文小结	13
参考文献	13

一 论文的基本介绍

之前的渲染技术对一些特殊场景，例如间接光照占主体（比如相机和光源之间有小隔板，光反射后照亮整间屋子）或者场景中都是很光泽的物体时，渲染效率并不会很高。老的扫描线或光线追踪方法也无法得到间接光照的效果。

研究光传输不但可以得到更好的渲染外观，光传输理论的结构类似于辐射和粒子传输问题，也会对这些物理或工程学起到促进作用。

在该文章中，光都是遵循几何光学模型的——光只在表面被发射、散射或者被表面吸收，在表面之间沿直线传播；该论文也忽略光的波动性或量子模型，只考虑几何光学。该论文主要有三个创新内容：新的理论模型、新的统计学模型以及新的渲染算法。下面三个子章节就介绍了主要创新点和贡献，这些内容可以只做提前了解，后面会有详细解释。

1.1 双向传输理论

本文提出了一个基于线性算子理论的简单的光传输模型，这个新的公式把光传输、重要性传输和粒子追踪进行了统一，并总结了它们之间的关系（本文不做一些场景的物理有效性假设，因此它是更全面更丰富的）。

有不少 BSDF 模型并不是对称的，例如由折射或者着色法向量引起的这种不对称性。本文推导了要在这些例子中使用双向路径追踪方法所需要的一些转换关系（如果不正确处理，就会出现大量伪影和错误）。

物理材质的光反射由对称 BSDF 描述。而本文导出了适用于任意材料（即透射和反射）的该条件的推广。该论文利用热力学定律，特别是基尔霍夫定律 (Kirchhoff's laws) 和详细平衡 (detailed balance) 原理，建立了新的互易原则 (reciprocity principles)，还讨论了互易原则的历史渊源、其论证中涉及的微妙之处以及它们有效的条件。利用这一新的互易原理，该论文提出了第一个光传输公式，即自伴随（共轭）算子 (self-adjoint operator) 公式。其中线性算子对于所有物理有效场景都是自共轭（对称）的，本文表明这简化了双向光传输算法的理论和实现。。

路径积分公式。通常，光传输问题用积分方程或线性算子表示。而该论文展示了如何将其表述为路径空间上的积分问题。这种观点允许应用新的解决方案技术，如多重重要性采样 MIS 或 Metropolis 光传输算法。

本文还表明某些类型的传输路径不能通过标准采样技术生成，这意味着无偏蒙特卡洛算法（如路径跟踪）生成的图像可能会丢失某些照明效果（无偏蒙特卡洛方法的固有局限性）。该论文分析了发生这种情况的条件，并提出了实现这些路径采样算法的方法。

1.2 更好的蒙特卡洛技术

多重重要性采样 MIS 可以更鲁棒的组合多种采样技术。我们可以同时使用两种采样技术来采样（比如同时采样光源和 BSDF），然后两个的结果取平均。但是直接取平均并不是最好的方法，MIS 可以更好地组合采样策略。MIS 也可以很好地应用到其他更多技术，比如双向路径追踪 BDPT。

俄罗斯轮盘可以降低采样的消耗时间，但是会增大方差。该论文提出了一种新的优化方法，将一个属性与另一个属性进行权衡，以最大化结果估计器的效率。这在可见性测试的上下文中特别有用，因为通常有许多样本只做出很小的贡献。

1.3 鲁棒的光传输算法

双向路径追踪方法。该论文提出了一种新的光传输算法，其思想是对路径使用一系列不同的采样技术，然后使用多重重要性采样将它们结合起来。每个路径通过连接两个独立生成的子路径生成，一个从光源开始，另一个从眼睛（相机）开始。通过改变光子路径和相机子路径的长度，获得了一系列不同的采样技术（注意这里“不同的采样技术”要理解为不同的采样路径长度方案），该论文表明，每种技术都可以有效地采样不同类型的路径，并且这些路径负责最终图像中的不同照明效果。通过使用多重重要性采样组合所有这些技术的样本，可以有效地处理各种不同的照明效果。

该论文描述了一套完整的双向估计器，包括光线或相机子路径最多有一个顶点的重要特殊情况。该论文还讨论了处理理想镜面、任意路径长度和有效可见性测试的扩展。

Metropolis 光传输。受计算物理中的 Metropolis 采样方法的启发，该论文提出了一种新的蒙特卡洛方法来解决光传输问题。为了渲染图像，我们通过随机突变单个当前路径来生成光传输路径序列（例如，突变可能会为路径添加新顶点）。每个突变都以精心选择的概率被接受或拒绝，以确保能够根据路径对所需最终图像的贡献对路径进行采样。以这种方式，我们在传输路径的空间上构造了一个随机游走 (random walk) 方法，从而可以通过简单地记录图像平面上这些路径的位置来形成无偏图像。

该算法是无偏的，处理一般的几何和散射模型，使用很少的存储，并且比以前的无偏方法效率高出几个数量级。它在通常被认为困难的问题上表现得特别好，例如涉及明亮间接光、小的几何孔洞或光滑表面的问题。此外，即使在具有相对简单照明的场景中，它也可以媲美先前的无偏算法。

Metropolis 方法的关键优势在于，路径空间是局部探索的，有利于对当前路径进行微小变化的突变。这有几个后果。首先，每个样本的平均成本很小（每个像素通常只发射一个或两个光线）。第二，一旦找到重要路径，也会探索附近的路径，从而在多个样本中分摊寻找此类路径的费用。第三，突变集易于扩展，通过构造在改变其他光源的同时保持路径某些属性（例如使用哪个光源）的突变，我们可以利用场景中的各种相干性。通过以这种方式设计专门的突变，通常可以有效地处理困难的照明问题。据我们所知，这是 Metropolis 算法首次应用于光传输问题。

二 光传输算法

光传输算法一般可以分为蒙特卡洛方法和有限元方法。Appel 于 1968 年开始在计算机图形学中应用蒙特卡洛方法；Whitted 于 1980 年开始引入光线追踪，还提出了随机扰动 viewing ray 的想法（在一个像素内扰动，可以抗锯齿）；Cook 于 1984 年扩展到对光源、镜头和时间（比如运动模糊）的采样。这些早期工作，使得 1986 年 Kajiya 得到了第一个无偏的完整蒙特卡洛传输算法，表明光传输问题可以写为积分方程，并通过采样路径来求解积分（从此，路径追踪诞生了）。从此，路径追踪技术的许多改进都是从粒子传输文献中改编的。

也有大量关于有偏蒙特卡洛算法的工作，这些算法通常比路径跟踪更有效。比如 1988 年 Ward 等人的辐照度缓存 (irradiance caching) 算法、Shirley 等人于 1995 的密度估计 (density estimation) 方法和 Jensen 于 1995 的光子映射 (photon map) 方法。

光传输的有限元方法最初改编自辐射热传输 (radiative heat transfer) 相关文献。Goral 等人于 1984 将这些方法引入图形社区，通常称为光能传递算法 (radiosity algorithms)。后来人们对基本辐射度方法进行了许多改进，包括 Cohen 等人 1986 年的子结构 (substructuring)、Cohn 等人于 1988 年的渐进细化 (progressive refinement)、Hanrahan 等人于 1991 年的层次基函数 (hierarchical basis functions)、Smits 等人于 1992 年的重要性驱动细化 (importance-driven refinement)、Lischinski 等人于 1992 的不连续网格划分 (discontinuity meshing)、Gortler 等人于 1993 年的小波方法和 Smits 等人于 1994 年的聚类 (clustering) 方法。其他扩展包括参与介质的处理以及非漫反射 (non-diffuse) 表面的有限元方法。

人们还提出了结合蒙特卡洛和有限元方法的方法，通常这些方法采用多步骤 (multi-pass) 方法的形式，将辐射度 (radiosity) 和光线跟踪步骤结合起来，以处理更一般的场景模型。另一种方法是蒙特卡洛辐射度，这里的解表示为基函数的线性组合（与有限元方法一样），但其中系数是通过跟踪随机光粒子 (random light particles) 来估计的。

2.1 蒙特卡洛方法与确定性方法的对比

在最基本的层面上，蒙特卡洛算法使用随机数，而确定性算法不使用随机数。然而，在实践中，算法通常使用混合技术，并且不容易分类。

准确性方法一般都会依赖于场景的显式表示（被场景表示的尺寸和复杂性强烈影响着）。蒙特卡洛算法基于采样，这意味着场景模型通过一小组查询（例如，给定光线相交的第一个曲面点是什么）访问。该接口将场景复杂性隐藏在抽象层后面，这意味着渲染时间仅与场景表示松散耦合（例如，场景复杂性可能

会影响光线求交所需的时间)。事实上,蒙特卡洛算法可以对场景进行采样,以确定它们实际需要的信息,而大多数确定性算法设计用于检查每个细节(无论这些细节是否相关)。

对于鲁棒性来说,这是一个特别重要的问题:理想情况下,光传输算法的性能应仅取决于场景所表示的内容,而不是如何表示的细节。例如,考虑由正方形区域光源照明的场景,如果将该光源替换为 10×10 的点光源网格,则视觉结果将几乎相同,然而,在点光源网格情况下,许多光传输算法的性能将更差。类似地,假设我们用一对覆盖有半透明面板的荧光灯泡替换同一光源。在这种情况下,整个场景被间接照明,这将导致许多算法出现问题,理想情况下,渲染算法不应对此类外观变化敏感。对几何复杂性也是如此:无论对象表示为一千个多边形还是一百万个贝塞尔面片,我们都希望渲染时间尽可能相似。蒙特卡洛算法至少有可能有效地处理这些情况,因为它们是基于采样的。

蒙特卡洛和确定性方法之间的区别有些模糊,因为蒙特卡洛算法对它们使用的数的“随机性”施加了非常弱的限制(例如,通常唯一的要求是这些随机数是均匀分布的)。通常可以设计满足相同限制(比如要求均匀分布这个限制)的固定采样模式,这通常会导致更好的性能(称为准蒙特卡洛方法)。蒙特卡洛方法的原理不是说样本是真正随机的,而是可以用随机样本代替它们。

2.2 视角相关与视角无关算法

视角无关 (view-independent) 算法是一种计算解决方案的中间表示的算法,可以快速生成任意视图;其他算法都是视角相关 (view-dependent) 算法。

对于动画来说,视角相关算法很有用,场景模型在一帧到下一帧之间会发生很大改变。视角无关算法对于交互应用尤其是游戏来说很有用。视角无关算法不保证在任意视角角度时的正确性,所以可能在某些位置出现错误的结果。场景中存在光泽物体时,则需要额外的工作来存储视角无关算法的一些中间结果。

即使只允许漫反射曲面,视角相关算法通常更高效,因为它们只需要计算我们感兴趣的解的部分。例如,如果场景模型很复杂,并且只有一小部分是可见的,那么直接计算图像会更有效。图像空间算法在这里具有最大的潜力,因为重要性驱动 (importance-driven) 的方法不能很好地扩展到复杂场景(在复杂场景中,计算整个域上的粗略解可能非常昂贵)。

视角相关算法和视角无关算法之间的差异实际上并不像最初看起来的那么大,因为通常可以将视觉相关算法转换为视觉无关算法。相似之处在于,两种算法都计算全局解的有限组线性度量 (linear measurements)。对于与视角相关的算法,这些测量值是像素值:每个像素通过积分落在图像平面的一个小区域上的光来定义。这与视角无关的方法密切相关,其中解决方案通常表示为基函数的线性组合(包括 21 世纪才出现的球谐光照)。视角相关算法通常可以用于估计这些基函数的系数,而不是图像的像素值,因为它们都被定义为线性测量(比如从视角采样来估计球谐光照的球谐系数)。

2.3 无偏与一致蒙特卡洛算法

通过一系列符合概率密度 p 的随机数 (X_1, X_2, \dots, X_N) 来估计一个函数 F (用 N 个随机数来估计表示为 F_N), 得到估计值 Q , 则误差为 $F_N - Q$, 设误差的期望为 $\beta[F_N] = E[F_N - Q]$, 叫做偏差 (bias)。

当 $N \rightarrow \infty$, $\beta[F_N] \rightarrow 0$ 则该方法是无偏的 (unbiased)。

直观地说,无偏估计器通过把结果平均来计算正确答案。就平均而言,有偏估计器计算的结果是错误的。然而,如果有偏估计量是一致的 (consistent), 则可以通过增加样本量使平均误差任意小。

我们认为,为了使光传输计算具有鲁棒性,无偏估计器是必不可少的,这一点很重要因为图形中使用的许多算法只是一致的(比如光子映射,理论上我们增加光子量就能使得误差的平均值越小,除非光子量无穷,否则并不能消除误差)。首选无偏算法的基本原因是它们使估计求得的解中的误差更容易,为了对计算结果更有信心,我们必须对该误差有一些估计。对于无偏算法,这简单地涉及计算样本方差,因为任何误差都会显示为样本之间的随机变化(对于仅仅是一致的算法,我们还必须限制偏差,一般来说这很难做到)。偏差导致的结果不是嘈杂的噪声,而是不正确的结果,通常在视觉上很明显,表现为不连续性、过度模糊或令人不舒服的表面着色效果。

无偏算法通常用于生成参考图像,用来与其他渲染算法进行比较。无偏方法对可能发生的误差类型提供了强有力的保证,它们可用于检测和测量近似引入的伪影 (artifacts)。对于具有真实复杂性的场景,无

偏算法是生成我们可以说是正确的图像的唯一实用方法。在其他条件相同的情况下，我们显然应该选择无偏算法。图形学中，无偏见的方法“计算过于昂贵”，通过近似可以在较短时间内获得可接受的图像。虽然在图形中的光传输算法方面已经有了大量的工作，但很少有针对无偏算法的工作。我们认为，在我们能够判断它们的能力之前，需要进行更多的研究。本论文的目标之一是探索无偏方法可以实现什么和不能实现什么，以帮助解决这些问题。

三 光学模型与它们对图形学的影响

可以在许多抽象层次上研究光传输，从二维“flatland radiosity”到量子模拟。手头上有各种数学模型是很有用的，这样我们可以选择适合每个任务的最简单模型。正如我们将看到的，一些光学现象对算法设计有着深刻的影响，而另一些可以很容易地添加到算法框架中。正是这种关于模拟哪些效果的选择区分了不同类型的渲染算法，并将图形中的光传输与其他领域中的类似问题区分开来。

在以下小节中，我们总结了现实世界中发生的重要光学效应，并讨论了它们对光传输算法的影响。光学现象根据能够解释它们的最不复杂的光学理论（几何光学、波光学或量子光学）进行分组，这些理论都解释了观察到的光现象的不同方面。

虽然我们一般主要关注的只是几何光学的一小部分内容，但了解更广泛光学领域的内容也会有好处。

3.1 几何光学

几何光学可以描述很多光学现象，包括发光、漫反射、镜面反射、折射以及吸收。严格完整的几何光学过于复杂，我们会做一些假设来得到更简单和快速的结果。

参与介质往往被忽视。通常，光可以在三维介质（如雾或明胶）中发射、散射或吸收，通过忽略这些可能性，假设所有散射都发生在模型表面，也意味着当光在表面之间传播时没有能量损失。

几何光学还允许介质具有连续变化的折射率。例如，当空气被加热时，就会出现这种情况，从而导致闪烁的“海市蜃楼”效应。理论上，这种效应使光传输问题更加复杂，因为光束不再在表面之间直线传播。相反，它们遵循 Eiconal 方程描述的曲线轨迹，必须以小步积分来确定光束的路径。由于这种效果对于大多数图形模型并不重要，因此通常会被忽略。

另一个常见的假设是光是单色的（即它具有单一频率），这通常只是为了简化算法的描述。有些人提出对多频率光谱的操作通过一个通用接口进行处理，因此可以很容易地替换成不同的光谱表示。

折射计算时，折射率可能取决于入射光的频率，从而导致人们熟悉的彩虹效应，即色散。

3.2 波动光学

光也可以被视为电磁波，可以解释几何光学处理的所有现象，以及其他一些现象。并不总是需要模拟光的波模型来获得波效应，例如，偏振可以很容易地添加到基于几何光学的渲染系统中。事实上，图形中的光传输模型通常结合了所有三种光学理论（几何光学、波动光学和量子光学）的特征。

波表现出的一种效应是衍射，它使光线在障碍物周围略微“弯曲”。虽然衍射在人体这种级别的尺度上一般都不明显，但对于小物体（例如，小于 10 个波长的物体）来说，衍射是不能忽略的，但很难将衍射纳入大多数光传输算法，因为它违反了光以直线传播的假设。

另一个重要的波效应是相干性。我们一直假设光波是完全不相干的，这意味着任何两个这样的波都没有相位相关性。非相干光束最重要的特性是，当它们叠加时，它们的强度线性相加（其中强度表示为均方振幅）。这与我们通常的直觉一致，例如，两个 100 瓦灯泡的亮度是一个 100 瓦的灯泡的两倍。

如果相位之间存在正相关，则称为相长干涉，否则称为相消干涉。当两个强度相等的相干光束组合时，产生的强度可以是 0 到 4 倍，这种效应是经典“双缝实验”中的亮带和暗带的原因。

3.3 量子光学

量子物理为光的行为提供了最详细、最精确的模型。这些效应中的一些没有用几何或波动理论来解释，但仍然与计算机图形学有关。这些效应之一是荧光：当光子被分子吸收，然后新的光子以不同的波长发射时，就会发生这种情况。这种效果在现实世界中非常普遍。例如，商业上使用荧光染料以获得更亮的颜色；这就是为什么衣服在紫外线下经常“在黑暗中发光”。通过允许不同波长的能量相互作用（以线性方式），荧光很容易添加到渲染系统中。

另一个有趣的效应是磷光，在这里，光子被吸收，并在稍后重新发射（通常在不同的波长）。这种效果对于大多数计算机图形应用并不重要；然而，在其他领域也存在类似的问题，这类时滞反应至关重要（例如放射性元素的衰变）。磷光的实现要求渲染算法随时间积分入射光，因为磷光表面的当前发射能量取决于其过去时间的曝光。

到目前为止，我们描述的所有效应都属于线性光学。考虑一个任意的光学系统，它将一束光束作为输入，并产生另一束光束为输出。如果输出波是输入波的线性函数，则光学系统是线性的；如果我们叠加两个输入波，输出必须是单独使用每个波时得到的输出之和，这一特性适用于几乎所有光学系统。然而，随着激光的引入，已经发现了非线性效应。例如，当高强度激光穿过某些晶体时，离开晶体的光的频率是进入晶体的光频率的两倍，这被称为倍频，这在低强度光下不会发生，因此这是非线性的一个例子。

还有许多其他效应，其解释基于量子物理。例如，光电效应或观察到的黑体辐射光谱分布。激光的解释也依赖于量子物理。然而，这些效果与计算机图形学无关。我们不需要模拟黑体辐射以将其包括在场景模型中。类似地，狭义相对论和广义相对论对于所有实际目的都可以忽略（例如，光在引力场中的弯曲）。

四 蒙特卡洛方法

本文不会详细介绍蒙特卡洛方法，更详细的理论解释和证明可以在 Google 查阅。

1777 年，Comte de Buffon 做了一个实验，给定一组相隔为 L 的平行线，在上面仍一个长为 d 的针，让针的长度 $d > L$ ，得到针与一条线相交的概率密度为 $p = \frac{2L}{\pi d}$ ，后来 Laplace 指出这种方法可以用来估计 π 的值。

蒙特卡洛方法起源于二战后的 Los Alamos 国家实验室，用于设计热核武器。后来 Stanislaw Ulam 使用随机采样来模拟中子的飞行轨迹。

4.1 求积分的法则的收敛速度上限

求积分有很多种方法，叫做求积分法则 (quadrature rules)。大多数求积分法则都更适用于一维空间，在高维空间中效率并不高：例如张量积法则，随着积分维度上升，效率急速下降，这被叫做维数灾难 (curse of dimensionality)。

并不是所有求积分方法都需要张量积形式，但是，Bakhvalov 法则表明任意求积分法则的收敛速度都会被卡在一个上限，我们下面就是描述一下这个上限。

考虑一个积分形式（其中 Ω 是积分域，这个积分域不一定是一维的， $\Omega = [0, 1]^s$ ）：

$$I = \int_{\Omega} f(x) d\mu(x)$$
$$f : \Omega \rightarrow \mathbb{R} \tag{四.1}$$

$d\mu(x)$ 是测量函数（测度论，可以把 $\mu(x)$ 简单理解为积分变量）：

$$d\mu(x) = dx^1 \cdots dx^s$$
$$x = (x^1 \cdots dx^s) \in [0, 1]^s$$

注意这里 x 的每个分量的积分区间都是 $[0, 1]$ ，因此积分值可以如此估计（求平均值）， x_i 表示第 i 个 x 样本：

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N f(x_i) \tag{四.2}$$

根据 Bakhvalov 原理, 对于一个函数 $f \in C_M^r$, 估计误差为 (第一行式子表示求 r 次偏导, 需要 f 为 r 阶连续且偏导有界):

$$\begin{aligned} f \in C_M^r : \left| \frac{\partial^r f}{\partial (x^1)^{a_1} \dots \partial (x^s)^{a_s}} \right| \leq M \quad \sum a_i = r \\ |\hat{I}(f) - I(f)| > k \cdot N^{-r/s} \end{aligned} \quad (四.3)$$

其中, $k > 0$ 只是一个取决于 M 和 r 的常量。

4.2 概率理论基础

对于多维随机变量 (X^1, \dots, X^s) 的联合累积概率密度分布 CDF:

$$P(x^1, \dots, x^s) = Pr\{X^i \leq x^i \text{ for all } i = 1, \dots, s\} \quad (四.4)$$

联合概率密度函数以及与 CDF 的关系就是:

$$\begin{aligned} p(x^1, \dots, x^s) &= \frac{\partial^s P}{\partial x^1 \dots \partial x^s}(x^1, \dots, x^s) \\ Pr\{x \in D\} &= \int_D p(x^1, \dots, x^s) dx^1 \dots dx^s \end{aligned} \quad (四.5)$$

其中 D 是勒贝格可测子集 (Lebesgue measurable subset), $D \subset \mathbb{R}^s$ 。

更一般地, 对于具有任意域中的值的随机变量 X , 其概率度量 (也称为概率分布或分布) 是一个度量函数 P (再解释一下, 大写的 X 表示具体的 x 变量):

$$P(D) = Pr\{X \in D\} \quad (四.6)$$

对于任意可测集 $D \subset \Omega$, 必须满足 $P(\Omega) = 1$ (也就是说 Ω 表示了整个事件空间)。由此可以再定义密度函数 $p(x)$ 和 $P(D)$:

$$\begin{aligned} p(x) &= \frac{dP}{d\mu}(x) \\ P(D) &= \int_D p(x) d\mu(x) \end{aligned} \quad (四.7)$$

由于密度函数依赖于测量 μ 来定义, 我们使用 $p = P_\mu$ 来表示对于特定 μ 的概率密度。以及定义一个符号 $u_x = \partial u / \partial x$ 经常被用来进行分析。当在同一个域上定义了多个相关的度量函数 (例如, 立体角和投影立体角度量, 将在后面描述) 时, 该符号将非常有用。

设一个变量 x 概率密度为 $p(x)$, 则对于随机变量 $Y = f(X)$, 期望是:

$$E[Y] = \int_{\Omega} f(x) p(x) d\mu(x) \quad (四.8)$$

对于一对随机变量 $X \in \Omega_1$ 和 $Y \in \Omega_2$:

$$(X, Y) \in \Omega \quad (四.9)$$

设 P 是联合概率测量, 对于任意可测子集 $D \subset \Omega$, $P(D)$ 表示 $(X, Y) \in D$ 的概率:

$$P(D) = \int_D p(x, y) d\mu_1(x) d\mu_2(y) \quad (四.10)$$

其中 μ_1 和 μ_2 各自是 Ω_1 和 Ω_2 上的测量。

定义条件概率密度函数 $p(y|x)$ 和 $p(x|y)$:

$$\begin{aligned} p(x) &= \int_{\Omega_2} p(x, y) dy & p(y) &= \int_{\Omega_1} p(x, y) dx \\ p(y|x) &= p(x, y) / p(x) \\ p(x, y) &= p(y|x) p(x) = p(x|y) p(y) \end{aligned}$$

对于随机变量 $G = g(X, Y)$, 定义:

$$E[G|x] = E_Y[G] = \int_{\Omega_2} g(x, y)p(y|x)dy = \frac{\int g(x, y)p(x, y)dy}{\int p(x, y)dy} \quad (四.11)$$

条件期望和方差:

$$V[G] = E_X[V_Y[G]] + V_X[E_Y[G]] \quad (四.12)$$

也就是说 $V[G]$ 是条件方差的均值加上条件均值的方差。

4.3 蒙特卡洛方法

我们假设读者具备蒙特卡洛方法的基础知识, 这里只是定义一些符号。

还是求积分问题:

$$I = \int_{\Omega} f(x)d\mu(x) \\ f: \Omega \rightarrow \mathbb{R} \quad (四.13)$$

设我们有符合概率密度 p 的样本 X_1, \dots, X_N , 则估计的积分值为:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} \quad (四.14)$$

下标 N 表示使用 N 个样本。

对于 $\Omega = [0, 1]^s$ 以及样本 X_i 都是均匀采样得到的且相互独立, 则:

$$F_N = \frac{1}{N} \sum_{i=1}^N f(X_i) \quad (四.15)$$

蒙特卡洛估计器是无偏的, 其实很好证明:

$$E[F_N] = E\left[\frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}\right] \\ = \frac{1}{N} \sum_{i=1}^N \int_{\Omega} \frac{f(x)}{p(x)} p(x) d\mu(x) \\ = \int_{\Omega} f(x) d\mu(x) \\ = I \quad (四.16)$$

蒙特卡洛积分的好处:

- 无论维度如何, 无论被积函数的连续性如何 (包括多维不连续函数), 其收敛速度都是 $O(N^{-1/2})$ 。
- 方法简单, 主要包含两个步骤: 采样和点估计 (point evaluation)。在面向对象的黑箱系统中也很好用 (比如包含运动模糊、参与介质的场景, 我们不知道积分估计会遇到什么, 但可以根据蒙特卡洛方法来计算积分结果)。
- 蒙特卡洛方法在路径空间积分中非常好用, 尤其是理论上光照路径是无限维的, 数值积分方法就很难处理这种情况。
- 最后, 对于具有奇点 (singularities) 的被积函数, 蒙特卡洛方法比数值积分方法更适合。重要性抽样可用于有效处理此类被积函数, 即使在没有解析变换来消除奇异性的情况下 (比如后面会介绍的拒绝抽样 (rejection sampling) 和 Metropolis 方法的讨论)。

蒙特卡洛积分的收敛速度证明可以参考论文原文的 2.4.1 节。这一小节里还通过切比雪夫不等式来描述了绝对误差的概率边界, 由于都是大学概率论的基础知识, 而且直接阅读原文并不难理解, 所以我不再摘抄上来了。

4.4 采样随机变量

有许多采样随机变量的方法，比如逆函数法（transformation or inversion method），即如果想得到概率密度分布为 p 的随机样本，需要得到相应的累积概率密度 P ，使得 $X = P^{-1}(U)$ ，这里的 U 是一个均匀分布的随机数；对于多维情况，则需要根据雅克比行列式来转换得到，或者先对每个维度的条件概率分布（边缘概率分布）依次生成样本。

这种逆函数法的好处在于可以很容易进行分层采样（通过满足参数空间 $[0, 1]^s$ ），并把这些样本映射到 Ω 上（就是说比如在 $[0, 1]$ 区间划分为 100 个等间距的子区间，这些子区间分别做均匀采样，得到 100 个更均匀的随机样本）。另一个优点是，该技术具有计算每个样本的固定成本（这个成本也很好计算得到）。主要缺点是概率密度 p 必须是解析的（需要知道它的公式形式），但有些时候 p 我们难以得到解析式。且累积分布 P 最好具有解析的逆函数，因为数值求逆函数通常较慢。

第二种采样技术是接受-拒绝法 (rejection method)，即从一些比较容易计算的概率密度 q 中采样，要求 (M 是某个常数)：

$$p(x) \leq Ma(x) \quad (4.17)$$

之后根据 $q(x)$ 采样 X_i ，再根据均匀分布 $[0, 1]$ 采样一个样本 U_i 。如果 $U_i \leq \frac{p(X_i)}{Mq(X_i)}$ 则返回 X_i 。

接受拒绝法好处是可以用于任意概率密度分布的采样（包括不解析的概率密度），但它需要知道 p 的上确界，如果这个上确界并不足够紧凑，则会有大量样本被拒绝，使得采样到一个样本的效率变低。此外，它很难应用分层抽样方法，最好的近似是在随机向量 (X, U) 域上进行分层（通过分层的方法生成 X_i 样本，然后通过分层的方法得到 U_i 样本），分层抽样的结果不如逆函数法好。

第三种采样技术是 Metropolis 方法（也叫 Markov chain Monte Carlo），它对于在高维空间中采样任意概率密度分布都是很有用的，且概率密度函数不需要进行归一化。而它最主要的问题在于产生的样本之间并不相互独立，而是具有很高的相关性，因此，如果我们需要根据概率密度 p 采样一段很长的样本序列时，该方法非常有用（我们会在后面再介绍 Metropolis 方法）。

还有一些采样特殊分布的技术，例如均匀采样一些独立样本 U_1, \dots, U_k ，然后找到它的最大值 X ，则 X 的分布概率密度就是 $p(x) = kx^{k-1}$ ，这些小技巧都可以被用来采样许多种标准分布，例如正交分布。

4.5 估计器的性能

设一个估计器有这种形式：

$$F_N = F_N(X_1, \dots, X_N) \quad (4.18)$$

注意 X_i 并不一定是相互独立的，甚至可能会有不同的概率分布， F_N 就表示使用 N 个样本来进行估计。 N 叫做样本量 (sample size)，每个 X_i 被叫做一个观察 (observation)。

估计器的目的是要近似感兴趣的量 Q 值（估计量，estimand），定义误差 (error) 为 $F_N - Q$ ，误差的期望就是偏差 (bias)：

$$\beta[F_N] = E[F_N - Q] \quad (4.19)$$

如果 $\beta[F_N] = 0$ 则估计器就是无偏的，即 $E[F_N] = Q$ ($N \geq 1$)。

估计器的一致性，即 ($Pr\{\cdot\}$ 符号表示某种事件的概率)：

$$Pr\left\{\lim_{N \rightarrow \infty} F_N = Q\right\} = 1 \quad (4.20)$$

也就是说，对于一致性估计器来说，它的期望并不一定等于实际值，但样本量趋近于无穷大以后，结果会越来越接近于实际值（比如光子映射）。一致性估计器随着 N 增长，其偏差和方差都趋近于 0：

$$\lim_{N \rightarrow \infty} \beta[F_N] = \lim_{N \rightarrow \infty} V[F_N] = 0 \quad (4.21)$$

均方误差写为：

$$\begin{aligned} MSE[F] &= E[(F - Q)^2] \\ &= V[F] + (\beta[F])^2 \end{aligned} \quad (4.22)$$

也就是说要估计估计器的均方误差，我们需要知道偏差，由此我们需要知道估计量的额外信息。对于无偏估计器， $E[F] = Q$ ，偏差为 0，因此均方误差就是：

$$MSE[F] = V[F] \quad (四.23)$$

根据样本方差来估计总体方差，获得一系列样本 Y_1, \dots, Y_N ，下面就是对方差 $V[F_N]$ 的无偏估计：

$$\hat{V}[F_N] = \frac{1}{N-1} \left\{ \left(\frac{1}{N} \sum_{i=1}^N Y_i^2 \right) - \left(\frac{1}{N} \sum_{i=1}^N Y_i \right)^2 \right\} \quad (四.24)$$

至于为什么分母是 $N-1$ ，可以参考总体方差与样本方差之间的关系（概率论基础知识）。只要样本是相互独立的，则满足：

$$V[F_N] = \frac{V[F_1]}{N} \quad (四.25)$$

也就是说样本越多，估计器的方差就越小，然而，这也会将运行时间增加 N 倍。理想情况下，我们希望找到方差和运行时间都很小的估计器。通过蒙特卡洛估算器的效率总结了这种权衡：

$$\epsilon[F] = \frac{1}{V[F]T[F]} \quad (四.26)$$

因此，估计器越有效，则在固定的运行时间内可以获得越低的方差。

五 减少方差的方法

减少方差的方法用来提高估计器效率，大致分为四个部分：

- 对类似于被积函数的函数进行解析积分。
- 在积分域均匀地放置样本。
- 基于在采样中搜集到的信息来自适应地控制样本密度。
- 组合值相关的两个或多个估计器的样本。

5.1 解析积分 (Analytic integration)

解析积分大致有三种方法：使用期望值 (the use of expected values)、重要性采样 (importance sampling) 和控制变量 (control variates)。

5.1.1 使用期望值

也许减少方差的最明显的方法是通过积分域的一个或多个变量进行解析积分来减少样本空间的维数。这种想法通常被称为使用期望值或降低维度 (reducing the dimensionality)。具体而言，它把估计器 $F = f(X, Y)/p(X, Y)$ 替换为 $F' = f'(X)/p(X)$ ：

$$\begin{aligned} f'(x) &= \int f(x, y) dy \\ p(x) &= \int p(x, y) dy \end{aligned} \quad (五.1)$$

要产生符合概率密度 $p(x)$ 的样本其实并不难，只需要跟以前一样生成样本 (X, Y) 然后忽略 Y 就可以了。估计器 F' 就是 F 的条件期望值：

$$\begin{aligned} F' &= E_Y \left[\frac{f(X, Y)}{p(X, Y)} \right] \\ &= \int \frac{f(X, y)}{p(X, y)} p(y|X) dy \\ &= \int \frac{f(X, y)}{p(X, y)} \frac{p(X, y)}{\int p(X, y') dy'} dy \\ &= f(X)/p(X) \end{aligned} \quad (五.2)$$

这使得减少方差很容易分析，使用替换式得到：

$$\begin{aligned} V[F] &= E_X V_Y F + V_X E_Y F \\ F' &= E_Y F \\ \implies V[F] - V[F'] &= E_X V_Y F \end{aligned} \tag{五.3}$$

这个量总是非负的（因为是方差的期望，一定大于等于 0），并且表示由 Y 的随机采样引起的 F 的方差分量（正如我们理解的，多了一个维度，方差就多一部分）。使用期望值是首选的减少方差的技术，只要估计或采样解析积分量计算消耗不大即可。但是要注意，如果期望值仅用于较大的计算的一部分，则方差实际上可能会增加。

5 1.2 重要性采样

就是让随机变量 X 的概率密度函数 p 尽可能接近被积函数 f 。

最理想的情况当然是 $p(x) = cf(x)$ ， c 表示一个常量；但这种情况一般不可得，我们或许不知道被积函数是什么样子。我们可以将被积函数分段线性近似，对每一段使用重要性采样策略。

重要性采样可能是最重要的方差减少工具了。

5 1.3 控制变量法

控制变量法，就是找一个可以被解析积分的函数 $g(x)$ ， $g(x)$ 要尽量跟被积函数 $f(x)$ 比较相似，然后：

$$I = \int_{\Omega} g(x) d\mu(x) + \int_{\Omega} [f(x) - g(x)] d\mu(x) \tag{五.4}$$

估计式为：

$$F = \int_{\Omega} g(x) d\mu(x) + \frac{1}{N} \sum_{i=1}^N \frac{f(X_i) - g(X_i)}{p(X_i)} \tag{五.5}$$

如果后半部分的方差比直接对 $f(x)$ 积分进行估计的方差小，则控制变量法就是有效的。不过需要注意如果 g 正比于 p ，则两种估计器的方差其实是一样的。

控制变量法在图形学中的应用比较少。

5 2 均匀样本放置 (Uniform sample placement)

另一种方差减少方法是让样本分布更均匀，有下面几种技术：分层抽样 (stratified sampling)，拉丁超立方体抽样 (Latin hypercube sampling)，正交阵列抽样 (orthogonal array sampling) 和准蒙特卡洛方法 (quasi-Monte Carlo methods)。这些方法获得的随机样本可以通过重要性采样技术（比如逆函数法）映射为其他概率密度分布的随机样本。

5 2.1 Stratified sampling

分层抽样将域 Ω 下分为几个不重叠的子空间 $\Omega_1, \dots, \Omega_n$ ：

$$\cup_{i=1}^n \Omega_i = \Omega \tag{五.6}$$

每个区域 Ω_i 被叫做一个层 (stratum)。在每个层 Ω_i 内以概率密度 p 来固定采样 n_i 个样本。

为了简化起见，假设 $\Omega = [0, 1]^s$ ， p_i 是 Ω_i 上的常量函数（均匀概率密度），则：

$$\begin{aligned} F' &= \sum_{i=1}^n v_i \left(\frac{1}{n_i} \sum_{j=1}^{n_i} f(X_{i,j}) \right) \\ &= \sum_{i=1}^n v_i F_i \end{aligned} \tag{五.7}$$

其中 $v_i = \mu(\Omega_i)$ 是区域 Ω_i 的体积（这里的体积其实是“超体积”，并不一定是三维的，对于一维来说就是每个子区间的线段长度），每个 $X_{i,j}$ 都是从 p_i 分布中采样出来的独立样本。

上式，每个 F_i 都是对一个子空间的估计，将所有子空间的估计值加起来，就得到对 I 的估计 F' 。估计器的方差是（这个方差很容易理解）：

$$V[F'] = \sum_{i=1}^n \frac{v_i^2 \sigma_i^2}{n_i}$$

$$\sigma_i^2 = V[f(X_{i,j})] \quad (五.8)$$

为了去对比非分层方法，假设 $n_i = v_i N$ （ N 是总样本数），则上式就可以写为：

$$V[F'] = \frac{1}{N} \sum_{i=1}^n v_i \sigma_i^2$$

对应的非分层抽样估计器的方差是（其中， μ_i 是 f 在区域 Ω_i 的均值， I 是函数在整个域上的均值，推导放在了下面）：

$$V[F] = \frac{1}{N} \left[\sum_{i=1}^n v_i (\sigma_i^2 + (\mu_i - I)^2) \right] = \frac{1}{N} \left[\sum_{i=1}^n v_i \sigma_i^2 + \sum_{i=1}^n v_i (\mu_i - I)^2 \right] \quad (五.9)$$

上式的右半部分一定是大于 0 的，我们也可以看出，当层的均值与整体均值不同时，分层抽样才会比较有意义。当被积函数是一个单调递增函数，我们就可以直接划分为多个相邻的段，这样结果就会很好了。

上式的推导：假设一个 $[0, 1]^s$ 上的非分层样本首先以概率密度 v_i （注意在 $[0, 1]^s$ 域上这个体积值就是概率密度，无需归一化）选择一个随机的层 I_j ，然后再在 Ω_{I_j} 上随机选择一个 X_j （可以感觉 X_j 是以在 I_j 为条件的），因此利用前面描述过的恒等式：

$$V[G] = E_X V_Y G + V_X E_Y G \quad (五.10)$$

代入：

$$V_X E_Y G = \frac{1}{N} \sum_{i=1}^n v_i (\mu_i - I)^2$$

$$E_X V_Y G = \frac{1}{N} \sum_{i=1}^n v_i \sigma_i^2 \quad (五.11)$$

合并就能得到目标结果。

如果函数一阶导数有界，则分层抽样的方差以速率 $O(N^{-1-2/s})$ 衰减。

如果被积函数值中具有奇点或快速振荡（例如，具有精细细节的纹理），或者当维数过高的时候，分层抽样方法就会显得没那么好用了。

5.2.2 Latin hypercube sampling

该方法我一直没有接触过，也没有使用过，所以不进行讲解，有需要可以参考论文 [1] 第 2.6.2 节。

5.2.3 Orthogonal array sampling

该方法我一直没有接触过，也没有使用过，所以不进行讲解，有需要可以参考论文 [1] 第 2.6.3 节。

5.2.4 Quasi-Monte Carlo methods

准蒙特卡洛 (Quasi-Monte Carlo) 方法通过完全消除随机性，通过确定地选择样本的位置，尽可能均匀地分布样本。

在图形学中，准蒙特卡洛方法、低差异序列 (low-discrepancy sequence) 等词一般都是同一个意思，就是应用一些具体算法来生成更均匀的样本。虽然该方法非常重要，但很多时候我们确实也不需要掌握它的基本原理，而实现时经常会涉及比较复杂的位移反转等操作，因此我们使用时一般会从 github 或者一些专门的实用算法书籍中移植现有的代码。

5.3 自适应样本放置 (Adaptive sample placement)

第三种方差减少的方案是基于自适应控制样本密度的，在更有用的地方放置更多的样本。一共有两种方式，一是自适应采样，除非采取特殊预防措施，否则会引入偏差；二是两种相似的技术，俄罗斯轮盘 (Russian roulette) 或 splitting，这两种技术不会引入方差，在光传输中的应用非常广泛。

5.3.1 自适应采样

自适应采样 (adaptive sampling) 又叫顺序抽样 (sequential sampling)，在被积函数方差更大的地方采样更多的样本。该方法通过测试已有样本的方差，在方差比较大的位置放置更多的样本。

它和重要性采样很像，都是在“更重要”的区域采样更多的样本，但是有两点不同：一是重要性采样会在被积函数值更大的区域放置更多样本，而自适应采样是在方差更大的区域（或者我们更感兴趣的区域）放置更多样本（当然，很多情况下数值更大的区域方差会更大）；二是自适应采样的样本密度会在采样中随时变化，而重要性采样的样本密度是提取定义好的。

自适应采样会引入偏差 (bias)，有些技术可以改进；另外，在高维问题上自适应采样可能并不是很有效。

5.3.2 Russian roulette

如果一个估计器是许多项的组合：

$$F = F_1 + \dots + F_N \quad (五.12)$$

比如 F 表示从表面反射到观察方向的辐射度，而每个 F_i 表示特定的光源的贡献。有些时候有些部分的贡献非常小，而所有的 F_i 都计算的话计算会比较昂贵，因此俄罗斯轮盘会选择根据概率“跳过”一些估计（设 $[0, 1]$ 之间一个均匀随机数 U ）：

$$F'_i = \begin{cases} \frac{1}{q_i} F_i & U < q_i \\ 0 & otherwise \end{cases} \quad (五.13)$$

则此时对于每个 F_i 的估计是无偏的：

$$\begin{aligned} E[F'_i] &= q_i \cdot \frac{1}{q_i} E[F_i] + (1 - q_i) \cdot 0 \\ &= E[F_i] \end{aligned} \quad (五.14)$$

而且我们很容易就能感受到该方法确实会增加方差，因为又引入了随机概率，但是如果有些估计的内容贡献很小，它可以很好地提升效率。比如当在光线追踪中，反射好几次的对射入眼睛方向的整个贡献会非常小，而且做光线追踪判断求交和遮挡性等的计算消耗大，使用俄罗斯轮盘就可以选择是否再去跟踪这些光路，从而极大地提升效率。

5.3.3 splitting

俄罗斯轮盘与 splitting（大致意思就是分裂）方法很像，将一个估计器分裂为：

$$F'_i = \frac{1}{k} \sum_{j=1}^k F_{i,j} \quad (五.15)$$

其中 $F_{i,j}$ 是从 F_i 中得到的独立样本，贡献越大的 F_i 就会设置更大的 k 值。

实践上来说，就是比如我们每个像素只发射 5 条光线来抗锯齿，而每条光线采样 10 个面光源点来获得更好的软阴影（共采样 $5 + 5 \times 10$ 次），这样得到的总体图像结果会比每个像素发射 25 条光线，每条光线采样 1 次（共采样 $25 + 25 \times 1$ ）的效果更好（第二种方式只计算采样 50 次，第一种方式计算采样 55 次，次数不同但比较接近）。

5.4 相关的估计器 (Correlated estimators)

最后一种方差减少的方案是基于找到一个或更多的值相关的估计器，但是这些方案在图形学中并没有多少应用，而且我也没有怎么接触过。

5.4.1 Antithetic variates

反向变量 (Antithetic variates) 是找两个估计器 F_1 和 F_2 ，它们的值是负相关的，然后加起来。例如目标积分是 $\int_0^1 f(x)dx$ ，考虑估计器 (U 表示 $[0,1]$ 上均匀分布的随机数)：

$$F = (f(U) + f(1 - U))/2 \quad (五.16)$$

如果函数单调递增或者单调递减，则 $f(U)$ 和 $f(1 - U)$ 是负相关的，如果两个样本相互独立， F 将会有更低的方差。

该方法对平滑被积函数来说最有用，而图形学中，方差的来源基本上都是被积函数的不连续性和奇异性（比如由遮挡关系造成的不连续性），因此并不是很有用。

5.4.2 Regression methods

该方法我一直没有接触过，也没有使用过，所以不进行讲解，有需要可以参考论文 [1] 第 2.8.2 节。

六 本文小结

本文是对论文 [1] 的前两个章节的分析与描述。

这两个章节由于内容相对简单和直观，所以分析较少，只在原文描述感觉不是很容易理解的地方多做了一点说明。

参考文献

- [1] Veach E . Robust Monte Carlo Methods for Light Transport Simulation[J]. Ph.d.thesis Stanford University Department of Computer Science, 1998.