arXiv:2301.05125v1 [cs.GR] 12 Jan 2023

自适应动态全局光照

SAYANTAN DATTA，麦吉尔大学，加拿大
NEGAR GOLI，华为/AMD，加拿大JERRY ZHANG，华为，加拿大

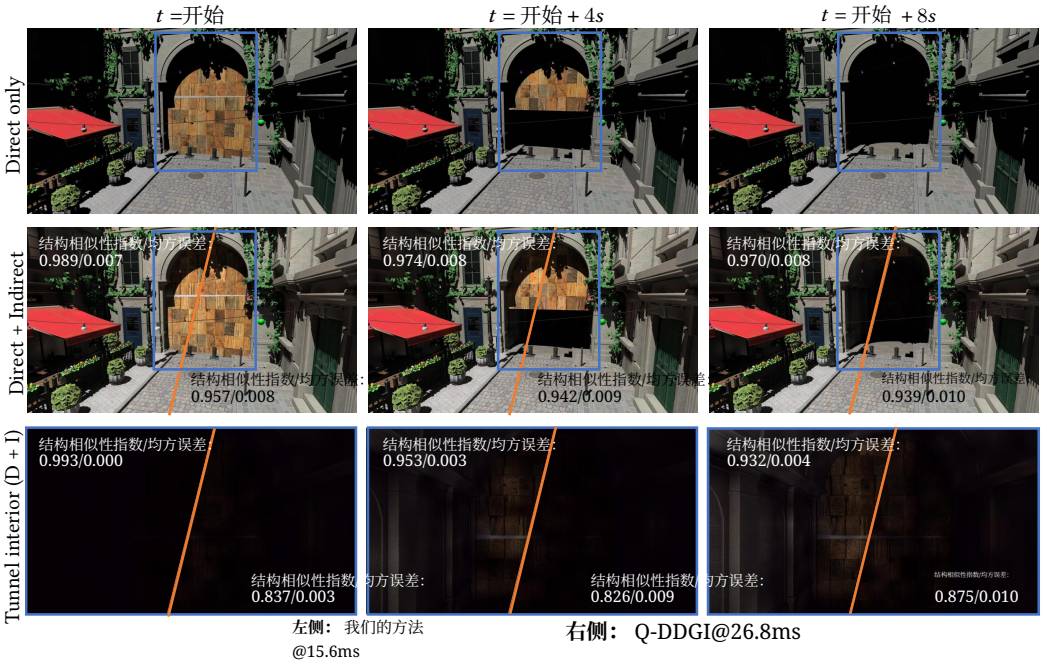


图 1. 我们的技术在一个经过修改的 Bistro Exterior 场景中展示，该场景包含 $192 \times 64 \times 192$ 个探针。第三行展示了随着闸门逐渐打开，隧道内部随时间发生的变化。即使探针网格规模庞大、包含超过 230 万个探针，我们的技术对动态刺激的响应速度仍然更快，性能也比 Q-DDGI 实现高出 1.7 倍。第 5 节将详细介绍 Q-DDGI。它是对原版 DDGI 的扩展，使其相较于我们的方法更具竞争力，也更可比。

我们提出了一种基于探针的全局光照解决方案的自适应扩展，可增强其对场景中动态变化的响应能力，同时还能将探针数量提高一个数量级。我们的自适应采样策略会谨慎地将采样点布置在检测到辐射度随时间变化的区域，这些变化可能由光照、几何体或两者的变化引起。即使探针数量庞大，我们的技术也能可靠地更新辐照度和可见性缓存，以反映最新变化，而不会使整个算法停滞。我们采用的带宽感知方法在很大程度上改进了原版动态漫反射全局光照，同时也与该技术近期的发展保持正交。[1] [2]

CCS 概念： • **计算方法** → 光线追踪； 光栅化。

附加关键词和短语：自适应采样、辐照度探针、全局光照、实时

1 引言

全局光照（GI）能够显著提升虚拟场景的真实感，但其高昂的计算成本长期以来一直是将其应用于实时渲染的难题 [22]。

[1]项目页面[2]海报

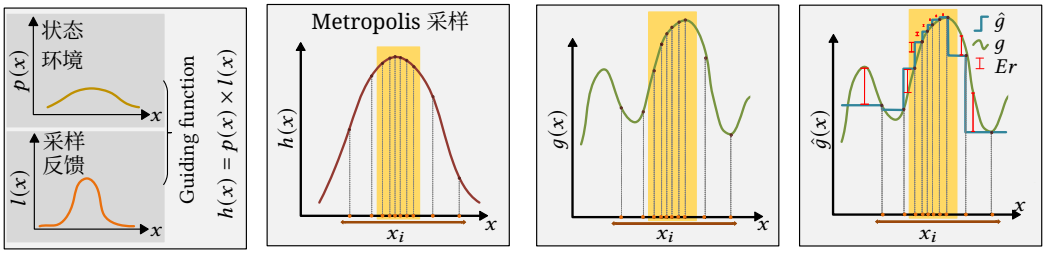
目前已有多种实时全局光照解决方案，例如屏幕空间 [43] 技术。这类技术支持完全动态的场景，但由于屏幕空间中可用信息有限，画质会受到影响。另一方面，烘焙纹理光照图仅支持静态几何体，但因其简单、运行时成本低且画质出色，至今仍广受欢迎。预计算辐射传输 [51] 结合光照探针 [31] 和光照图 [15]，解决了静态光照图面临的部分问题；特别是，这些方法在严格控制计算预算的同时，支持半动态几何体和自遮挡。实时光线追踪硬件的问世，为现代完全动态全局光照奠定了基础。动态实时全局光照方法建立在数十年的采样研究之上，通过在空间（像素/世界）、角度和时间维度上对着色与可见性进行摊销，提升收敛速度 [40]。人们还在现代 [26, 27] 支持光线追踪的硬件上，探索了多种离线技术的改进方案，例如光子映射 [17], 、多光源渲染 [20, 62], 和辐射度图 [54] 。然而，采样算法中存在的噪声要求使用强大的降噪器。与更传统的基于频率 [32] 或方差 [46] 的降噪器相比，机器学习降噪器 [6, 66] 在画质方面表现出显著优势。然而，训练神经网络的前景、将机器学习推理整合进传统图形管线所带来的复杂性，以及机器学习框架的专有性质，阻碍了这些技术在整个行业的普及。近期出现的基于探针的算法——动态漫反射全局光照（DDGI）[28], ——扩展了经典的辐照度探针。它兼具相对简单、画质出色和支持云流式传输等优点，至今仍是一个极佳的选择 [14, 53]。然而，DDGI 原始方案的扩展能力有限，因此需要采用多网格层次结构、探针滚动 [29] 等方法，使其能够扩展到大型环境。我们的自适应方法着重处理包含数百万个探针、且采用单一层次结构的环境中的动态内容。

我们提出了自适应动态全局光照（ADGI）算法，每帧追踪少量试探光线以扫描环境，并构建动态事件的粗略代表性模型。通过马尔可夫链采样，我们动态地将资源分配给关键区域，从而提高这些区域的收敛速度。DDGI 为每个探针分配固定数量的采样，并将采样均匀分布到各个方向；而 ADGI 则对探针所表示的离散化五维光场的联合空间角度域进行非均匀采样。我们的方法从根本上将资源分配与探针数量解耦，从而实现由用户控制的性能目标（每秒帧数），即使探针数量达到数百万，也能提升扩展性。此外，在渲染时间相同的情况下，我们的方法在静态和动态环境中都能更快收敛。我们的方法可直接兼容原始实现及其他多种扩展，例如探针滚动和探针体层次结构 [29]。

我们通过构建一种引导函数逼近技术来实现这些目标。该技术针对引导函数标出的特定区域进行有针对性的精确处理，从而无需进行均匀资源分配。此外，我们还开发了一种基于时间马尔可夫链的采样方法，它能够自然适应动态环境，同时支持扩展到大量探针。最后，我们讨论了专为此目的设计的颜色压缩方案，以节省内存和带宽。

2 相关工作

基于探针的方法：现代游戏广泛依赖光探针来实现静态和动态全局光照，因为这类方法易于集成到游戏引擎管线中，且运行时成本较低。一些人主张采用均匀网格探针布置，因为这种方法简单；另一些人则提出采用非均匀探针布置，以提高效率。基于探针的技术通常容易出现光泄漏。因此，均匀网格方法 [28, 31] 会在探针中存储额外信息，以判断探针对于某个着色点是否可见。非均匀方法则可能



(a) 构建引导函数 $h(x)$。 (b) 采样 $x_i \sim h(x)$。 (c) 评估目标函数 $g(x_i)$。 (d) $\hat{g}(x)$ - 重建 $g(x)$ 来自 $g(x_i)$。 Er 表示重建误差。

图 2. 展示我们自适应采样策略各步骤的示意图。我们定义了一个引导函数 $h(x)$，用于突出显示定义域中的重要区域（以黄色标示）。从 $h(x)$获得的采样 x_i 用于评估目标函数 $g(x_i)$。我们的目标是根据（x_i，$g(x_i)$)配对，获得 $g(x)$的近似表示，记作$\hat{g}(x)$。随着从突出显示的区域获取更多采样，如子图 (d) 所示，黄色区域的重建误差会降低。

采用精心设计的探针布置 [63]，并结合八叉树等空间数据结构，确定从表面元素观察探针的可见性。McGuire 等人 [28, 31] 存储探针周围几何体的深度值，并采用类似方差阴影映射[9] 的方法来近似计算可见性。然而，由于初始构建成本高，非均匀方法大多仅限于静态几何体。一些方法使用光栅化 [31, 63]，另一些则可能使用光线追踪 [28] 来计算探针内容。基于探针的技术在探针信息的存储方式上也有所不同：有些使用离散纹理 [28, 31]，另一些则可能采用球谐函数 [14, 55]等压缩基底表示。球谐函数会在存储前对内容进行隐式预滤波，但可能导致明暗振铃问题。读写探针所需的内存带宽也是一个主要问题。纹理压缩 [31, 53] 通常是减少内存带宽的首选方案。带宽对于探针数据的云流式传输也至关重要。在这类场景中，球谐函数[14] 表示可能更合适，因为它能很好地压缩低频数据。在运行时，动态基于探针的 [28] 全局光照解决方案会将光线均匀分配到各个探针，以更新其内容；随着探针数量增加，这很快就会成为瓶颈。相比之下，我们的方法着重于优化资源分配，以最大限度地提高视觉保真度。人们还提出了多种扩展方法，以提高均匀网格方法的可扩展性 [29]，例如多体积层级结构和探针滚动。我们的方法与这些扩展基本正交，并且完全兼容。

自适应采样：自适应采样已应用于屏幕空间光线追踪全局光照，在噪声较高、频率较高的区域累积更多采样 [12]。自适应采样也有助于过滤软阴影 [32],：先导光线对阴影半影的空间频率进行建模，并确定每个像素所需的额外采样数，以改善收敛。还有研究提出了自适应采样的神经网络版本 [13]，由神经网络生成采样图，并使其与后处理神经网络降噪器紧密耦合。从概念上说，我们的方法与之相似，但具体实现针对探针的时间相干采样问题进行了调整。有关辐照度缓存、屏幕空间全局光照和马尔可夫链蒙特卡罗技术的更多相关工作，请参阅第 8 节。

3 概述

我们重点关注 DDGI 原始方案中的两个主要问题。首先，该技术不支持非均匀分配资源，因此会在对视觉保真度并不关键的区域进行不必要的探针更新。其次，探针更新速度不够快，无法反映场景环境中的瞬态变化。我们的自适应策略会检测环境变化，并根据检测到的变化分配资源。尽管检测阶段需要分配额外资源，但实证评估表明，我们的非均匀自适应采样弥补了检测阶段损失的效率。检测阶段还使探针能够快速更新，从而捕捉场景中的瞬态变化。我们将该技术建模为引导函数逼近：利用引导函数驱动的离散表示（例如探针）来逼近连续函数（例如五维光场）。

一种近似连续函数的朴素方法是将定义域离散化，并为每个离散单元保留一个代表性样本。当定义域相对较小时，这种策略很有用；然而，随着定义域扩大或离散单元数量增加，在实时场景下更新所有离散单元的代价会高得难以承受。这是原始 DDGI 技术所面临的问题之一。在许多应用中，并不需要对整个定义域进行均匀更新；相反，我们可以容忍某些区域具有比其他区域更大的近似误差。一个简单的例子是注视点渲染，在这种情况下，周边区域的误差比注视中心附近的误差更不易察觉。在我们的场景中，对最终着色有贡献的探针需要最高的精度。

我们引入了引导函数这一概念，用于突出显示需要更高重建精度的区域。我们通过多个项的乘积来定义引导函数——第一项表示环境的当前状态，第二项则是来自采样缓存的反馈。我们使用具有时间连贯性的马尔可夫链对引导函数进行采样，并利用这些样本，通过并行线程安全方法更新我们的近似表示。因此，我们的方法可以概括为三个步骤——定义引导函数、对引导函数进行采样，以及使用样本更新近似表示。我们将在第 3.2、3.3 和 3.4 节介绍这些步骤，并在第 4 节讨论各种特定于实现的细节。请参见图 2。

与原始 DDGI 相比，我们的方法具有两项显著优势：近似质量和可扩展性。我们可以随时将资源集中在可能具有挑战性的区域，而不是整个定义域。只要引导函数能正确识别这些具有挑战性的区域，就能通过在适当区域投入更多资源来提升质量。由于我们对引导函数的采样独立于离散化数量，这种解耦使我们能够静态分配大量探针，而不影响运行时性能。增加离散化数量可以提高近似质量，而采样不受离散化数量影响则提升了可扩展性。更具体地说，我们可以在不影响性能的情况下，透明地增加离散探针的数量。运行时性能取决于生成的采样数量；引导函数会将采样导向适当区域。我们的马尔可夫链采样具有高度并行性、时间连贯性和可扩展性，因此适用于对大型探针网格进行实时的时间分布式重建。

3.1 背景

这里简要介绍原始 DDGI 算法。DDGI 由一个三维网格组成，其中包含方向分辨的辐照度探针，并通过硬件光线追踪实时更新。探针还包含可见性信息，以防止光泄漏。探针表示有诸多优点：它能很好地处理漫反射间接传输，而且对探针信息进行编码和解码的成本相对较低。该算法在每个活动探针处，以随机旋转螺旋模式从探针中心向外均匀分布射线采样。DDGI 是

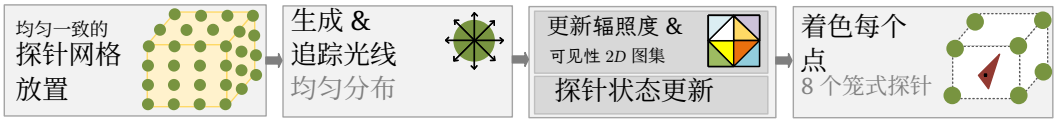


图 3. 该图展示了 DDGI 算法的主要步骤。算法定义一个探针均匀网格，并从每个探针沿所有方向追踪均匀随机射线。根据命中信息，我们计算可见性（vis）和辐照度（irr），并更新 $2D$ 图集。我们还根据可见性信息（背面命中率）更新探针状态。最后，对于每个着色点，我们查询其周围的八个包围探针，并通过插值计算入射间接照明。

一种两步算法。首先，它会更新探针纹素上的着色；接着执行屏幕空间遍历，使用最新的探针内容为相机像素着色。探针纹素值会编码到分辨率为 8×8 的球面映射漫反射辐照度纹理中。探针还会捕获射线命中距离的平均值，以及分辨率为 16×16 时到最近几何体的距离平方。DDGI 通过使用固定的滞后混合新值，对探针纹素进行时间滤波。可见性数据用于判断探针在着色点处是否可见，也用于推断探针是否位于几何体内部并将其停用。探针的状态并非仅限于开启或关闭，还会随场景 [29] 而变化。屏幕空间像素的世界位置用作探针纹理查找的键。查找会对包含着色点的网格体素中的相应八个探针进行插值。该算法如图 3 所示。DDGI 算法适用于漫反射以及随时间缓慢变化的现象。因此，DDGI 与我们的自适应采样策略相结合，是一种适用于动态场景的合理实时全局光照近似。

3.2 引导函数

如第 3 节和图 2 所述，引导函数 会突出定义域中的重要区域，即需要更多资源的复杂区域。这些突出显示的区域会获得更多自适应样本，从而降低这些区域的近似误差。从数学上讲，引导函数 h 的定义域为： $R^d \to R$ 连续的五维光场。查询时，引导函数会返回一个标量值，用以指示某个采样点的重要性。在我们的情形中，$d = 5$由于定义域是由世界空间位置和方向构成的五维空间，因此引导函数编码了对探针上的某个方向进行采样的重要性（纹素重要性）。

我们将引导函数（h）建模为两个项的乘积。第一项称为 f： $R^d \to R$，它表示根据我们对某个纹素是否会对最终屏幕空间着色产生贡献的有限认知，采样该纹素的价值。第二项是观测到的采样证据（又称辐照度缓存），随着这些证据逐渐可用而加入。起初，辐照度缓存为空，但会通过采样逐步填充。我们根据一些启发式方法定义第一项，以描述我们对探针环境的认知：

- 更靠近相机的探针，
- 更靠近几何表面的探针，
- 探针上背离几何表面的方向，
- 探针上入射辐照度较高的方向，
- 辐照度和可见性随时间发生变化的方向

我们从探针发射试探光线，以生成量化上述启发式方法所需的信息。我们也将其称为检测阶段，在此阶段，我们会预扫描场景环境以检测变化。我们将各个启发式方法记为 f_i： $R^d \to R$，并将它们组合成最终形式 f，如公式 1 所示，其中 ϕ 表示复合函数。复合函数是

算法 1： Metropolis 算法

输入： h：引导分布，M：迭代次数
输入： K：要舍弃的初始样本数
输出： x：采样
确保： $M \geq 2$，并且 $K < M$

```
j ←ShaderInvocationIndex()
x_0 ← S[j]                                   // 从内存初始化马尔可夫链
当 i ← 0 到 M − 1 执行
    x_{i+1} ←随机游走 (x_i, h(x_i))           // 随机游走步，算法 5
    如果i > K 则
        /* 使用采样 x_{i+1} 来更新探针，参见算法2 */
S[j] ← x_i + 1                               // 保存马尔可夫链状态
```

一种用于恰当组合各个启发式方法的简单方案。我们在第 4.1 节量化各个启发式方法（f_i），并在第 4.2 节量化组合（ϕ）。

$$f = \phi(f_0, f_1, ..., f_i). \tag{1}$$

第二项使用探针中存储的辐照度（记为 $\hat{g}$）调节第一项。我们将第二项建模为 - $exp(\alpha \cdot \hat{g}(x)/f(x))$，其中标量 $\alpha \in [0, \infty)$表示我们对辐照度探针内容的置信度；数值越高，置信度越高。请注意，存储的高辐照度值纹素对最终着色的贡献不一定很大。例如，在动态环境中，上一帧的探针内容很快就会过时，因此作用也较小。参数 α 用于建模这种不确定性。分母中的项 $f(x)$确保只有当 $f(x)$较低时，我们才会信任 $\hat{g}(x)$。最后，我们将引导函数定义为：

$$h(x) = exp\left(\alpha \cdot \frac{\hat{g}(x)}{f(x)}\right) \cdot f(x). \tag{2}$$

3.3 对引导函数进行采样

接下来，我们对引导函数（公式2）进行采样。数学上，给定一个未归一化分布 h：$R^d \to R$，我们的目标是从 $h(x)$中获取样本 x_i，其中 $x_i \in R^d$。

我们的采样算法很简单。我们使用 Metropolis 采样，如算法 1 所示，对 h 进行采样。算法会随机初始化一个状态（$x_0 \in R^d$），然后根据新提议状态是否被接受来推进状态。我们通过对当前状态添加零均值高斯噪声来生成提议状态，这也称为随机游走 [5]。

并行性： 请注意，算法 1 作为一次着色器调用运行，这意味着链的多个实例会并行运行。每个实例彼此独立，并拥有自己的内存，用于加载和存储链状态（在算法 1 中记为 S []）。这些实例每帧会生成数千个采样。作为算法的输入，我们会明确指定并行运行的链数量，从而控制自适应样本的数量和性能。与原始 DDGI 相比，原始实现中的采样数量与探针数量成正比，而探针数量会随着场景尺寸呈立方增长。因此，当场景变大或使用更密集的探针网格时，扩展规模会很困难。我们的方法不受离散化分辨率影响，并且在不降低近似质量的情况下，能够更好地扩展到更高的探针数量。

混合时间： 起初，马尔可夫链需要经过许多次迭代，才能生成来自目标分布（此处为 $h(x)$）的采样，这种现象称为混合时间。我们通过

表 1. 符号列表

符号	说明	备注
f	启发式模型	第 3.2 节
h	引导函数/目标分布	第 3.2、3.3 节
g	目标函数	g_r、 g_c 的符号代理。第 3.4 节
$\hat{g}$	目标函数近似	$\hat{g}_r$、 $\hat{g}_c$. 的符号代理 第 3.4 节
g_r	五维光场	第 3.4 节
g_c	切比雪夫可见性	第 3.4 节
$\hat{g}_r$	五维光场的近似（辐照度缓存）	第 3.4、4.5 节
$\hat{g}_c$	切比雪夫可见性的近似（可见性缓存）	第 3.4 节、第 4.6 节
x 或 x_i	马尔可夫链样本	p_i、 ω_i 的符号代理。第 3.3 节
p_i	$\in R^3$的位置分量（ x_i）	–
ω_i	$\in R^2$的方向分量（ x_i）	–

从上一帧引导初始链状态，来避免这个问题。因此，我们将每帧的迭代次数保持在较低水平；但随着帧数增加，链实际上会累积许多次迭代。

分布平稳性：马尔可夫链采样要求目标分布 $h(x)$保持平稳。由于场景环境是动态的，平稳性条件似乎遭到了破坏。如果分布在帧与帧之间变化过快，这可能会影响我们这项技术的近似质量。不过，我们有多种应对措施。首先，我们的目标是实现较高的帧率，从而尽可能减小连续帧之间目标分布的变化。为进一步确保安全，如算法 1 第 5 行所示，我们会舍弃每帧最初的 K 个采样。这样可以确保我们使用的采样更接近目标分布。请注意， $h(x)$ 的求值时间可以忽略不计，因此每帧舍弃少量初始采样不会显著影响性能。我们还使用时空卷积对目标分布进行平滑处理（见第 4.1.4 节），以尽可能减小目标分布在帧与帧之间的突变。

时间跟踪：由于目标分布可能随时间变化，我们要求马尔可夫链生成的采样能够紧密跟随该分布，以捕捉环境中的瞬态变化。我们对采样算法进行了一些关键修改，使其能够快速跟踪目标分布，详情见第 4.9 节。

3.4 近似

利用从定义域中高亮显示的部分（图 2(b)）获取的采样，我们重点关注如何使用这些采样来评估（图 2(c)）并重建（图 2(d)）目标函数。目标函数指的是我们希望近似的量。数学上，我们将目标函数记为 $g\colon R^d \to R^c$，其近似重建记为 $\hat{g}$。对于 ADGI，我们有两个目标函数——光场 $g_r\colon R^5 \to R^3$，以及探针周围的切比雪夫可见性 $g_c\colon R^5 \to R^2$。我们分别将它们的近似重建记为辐照度缓存 $\hat{g}_r$ 和可见性缓存 - $\hat{g}_c$。更多详情请参见第 4.5 节和第 4.6 节。

更新 ˆg：我们在收集到的采样点 x_i 处评估连续目标函数 g，并将评估结果 - $g(x_i)$ 存储到 $\hat{g}$中，如算法2所示。对于 ADGI，评估步骤包括

算法2： 近似算法

输入： x：马尔可夫链样本

函数UpdateRepresentation(x)：
　$v \leftarrow g(x)$　　// 评估采样，光线追踪
　AtomicMovingAvg(x, v)　　// 填充 $\hat{g}$，参见算法 4

追踪一条射线，以查询局部光场和可见性。在每次Metropolis迭代中，评估后的采样会在临界区结构中更新探针 ($\hat{g}$) 中距离最近的条目。

表示 $\hat{g}$： 以往的工作将 $\hat{g}$ 表示为离散查找表 [28], 连续球谐函数 [14], 神经网络 [36], 或它们的任意组合。在我们的案例中，选择使用离散表示是基于多个因素。首先，多个并行的马尔可夫链样本流可能会更新 $\hat{g}$中的同一内存位置。因此，有必要采取措施防止竞态条件。我们还需要一种能够处理时间累积并快速更新自身以反映场景中任何瞬态变化的表示。最后，该表示必须具备高带宽效率，以提升读写性能。详情请参阅第 4.5 节和第 4.9 节。

3.5 MCMC分析

本节将在 MCMC（马尔可夫链蒙特卡罗）的背景下分析我们的自适应采样算法。需要注意的是，我们的目标并非通过重要性采样来进行方差缩减；相反，我们关注的是通过对目标函数进行采样来实现目标函数的引导函数逼近。因此，与重要性采样不同，采样函数不一定与被积函数相关。明确这一点后，我们首先考察使用 MCMC 驱动重要性采样的方程，然后将其重新用于引导函数逼近。

下式展示了重要性采样的一个典型情形：目标是计算积分 $\int h(x)g(x)dx$，并且存在一种从 h(x) 进行采样的策略。在许多典型场景中（例如完整的贝叶斯推断），分布 $h(x)$ 是一个有效分布（$\int h(x)dx = 1$），但没有高效的采样机制。这正是马尔可夫链蒙特卡洛发挥作用的地方。

$$\int h(x)g(x)dx \approx \left\{\frac{1}{M}\sum_{i=0}^{M-1} g(x_i)\right\} \int h(x)dx, \quad x_i \sim h(x). \tag{3}$$

相比之下，我们选择马尔可夫链（Metropolis）主要出于技术方面的考虑——简单性、GPU 并行以及时间样本跟踪。尽管如此，同样的方程仍能提供有意义的洞见，只是应用于自适应采样的不同语境。在我们的算法中，我们只是将从目标分布中获得的采样相加，而不考虑样本密度。这相当于计算以下内容：

$$I = \frac{1}{M}\sum_{i=0}^{M-1} g(x_i), \quad x_i \sim h(x). \tag{4}$$

虽然我们的目标是估计 $\int_\Omega g(x)dx$，但期望值为 I（重排方程 3）i

W　　s:

$$\mathbb{E}[I] = \frac{\int_\Omega h(x)g(x)dx}{\int_\Omega h(x)dx}, \tag{5}$$

其中，Ω 是积分域。显然，I的期望值不会收敛到正确的估计值—— $\int_\Omega g(x)dx$。不过，有两个因素需要考虑：定义域 Ω 的大小，以及

定义域内 $h(x)$的形状。首先考虑 $\Omega \to 0$的极限情况。在这种情况下，积分会收敛为对一个点的求值，而 I的期望值确实等于如下所示的无偏估计值。

$$L.H.S. = \lim_{\Omega \to 0} \frac{\int_\Omega h(x)g(x)dx}{\int_\Omega h(x)dx} = \frac{\int_\Omega h(x)g(x)\delta(x-x_0)dx}{\int_\Omega h(x)\delta(x-x_0)dx} = g(x_0). \tag{6}$$

$$R.H.S. = \lim_{\Omega \to 0} \int_\Omega g(x)dx = \int_\Omega g(x)\delta(x-x_0)dx = g(x_0). \tag{7}$$

在上述公式中，δ是克罗内克δ。这个结果很重要，因为它表明随着探针分辨率提高，偏差会减小。然而，减小纹素大小并不总是可行的，因为填充和存储高分辨率探针需要更多光线和内存。注意，公式 6 中的项 $h(x)$被约去了。当积分域足够小时，$h(x)$实际上是常数，因此该项在分母和分子中相互抵消。

现在我们来考察 $h(x)$的形状。虽然目标 h(x) 在全局范围内变化，但由于其表格特性，在局部尺度上是分段常数。更重要的是，与辐照度探针 $\hat{g}(x)$ 相比，目标 $h(x)$的存储分辨率低得多。这意味着 $h(x)$在辐照度探针的一个纹素范围内实际上是常数。因此，k^{th} 纹素的 I的期望值为：

$$\mathbb{E}\left[I_k\right] = \frac{\int_{T_k} h(x)g(x)dx}{\int_{T_k} h(x)dx} = \frac{\int_{T_k} c_k g(x)dx}{\int_{T_k} c_k dx} = \frac{\int_{T_k} g(x)dx}{\int_{T_k} dx}, \tag{8}$$

其中，T_k表示 k^{th} 纹素的定义域，c_k表示当 $x \in T_k$时$h(x)$的分段常数值。面积估计值 $\int_{T_k} dx$对所有纹素均固定，且等于 $4\pi/\#resolution$。因此，由于我们的目标函数具有表格形式，辐照度纹素的估计值仍然无偏。在对辐照度纹素进行纹理过滤时，可以按如下方式使用 c_k对纹素值加权，以计算无偏估计：

$$I_k^{filter} = \sum_{j \in \mathcal{N}_k} w_{k-j} I_{k-j}, \quad w_i = c_i / \sum_{j \in \mathcal{N}_k} c_j, \tag{9}$$

其中，N_k表示纹素 k邻域内的纹素。数值 c_i通过查询存储 $h(x)$的探针获得。请注意，在动态环境中对采样进行时间混合时，偏差不可避免。在动态环境中，目标函数不断变化，偏差表现为时间滞后。但在实际应用中，在较短的时间窗口内，假设 $h(t)$和$g(t)$均为常数，并且可以使用窗口移动平均对采样进行混合。请注意，窗口移动平均需要存储历史信息。指数移动平均是一种成本更低但有偏的窗口移动平均近似方法。

4 实现细节

本 节介绍了若干实现细节，并在图中作了简要总结 4.

4.1 启发式信息构建

本节介绍如何使用第 3.2 节讨论的启发式方法构建 f。我们的目标是衡量并量化这些启发式方法，以突出显示那些对最终着色有积极贡献、且需要额外资源以加快收敛的探针。我们用参数化形式（公式 10）或显式查找表表示形式（如图 5(a) 所示）来表示这些启发式方法。构建查找表时，每个探针对应八个纹素，分别对应一个八分体。我们为每个八分体追踪一条射线；这些射线会返回命中距离以及命中点处的入射辐照度。根据这些信息，我们计算若干量（公式 11–18），并将其存储在

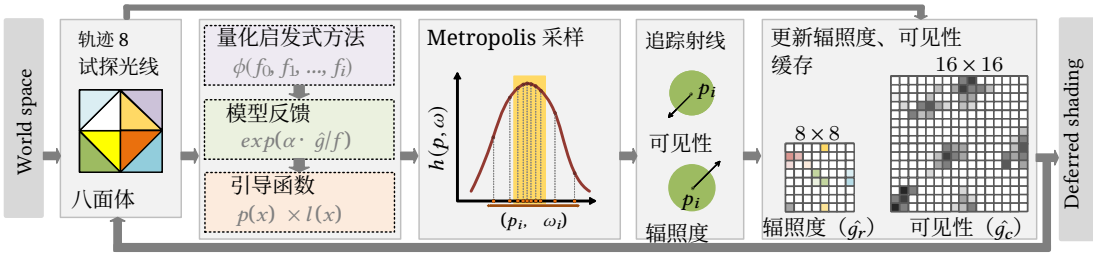


图4. 该图展示了我们的整体算法。我们从探针的每个八分体各追踪一条试探光线，共8条，并近似计算启发式模型 $f(p, \omega)$。利用启发式模型和反馈，我们定义引导函数$h(p, \omega)$，并使用 Metropolis 采样对其进行采样。采样得到的 (p_i, ω_i) 用于追踪更多自适应射线样本，从而在采样点处收集命中距离和辐照度。我们使用自适应采样更新探针缓存（$\hat{g}$）。该缓存用于下一个着色器，同时也作为反馈回传，以对目标进行建模。

映射到探针八分体的查找表/纹理中。对于位置为 p、方向为 ω的探针，我们定义并评估以下启发式方法。

4.1.1 与相机的距离。距离相机较远的探针对最终着色的贡献可能性较低。我们按照公式10所述，用参数化方式表示这一点，其中 p 表示探针位置，c 表示相机位置，k 是用户设定的阈值。

$$f_c(p, \omega) = \begin{cases} 1 & \text{if } ||p - c|| < k, \\ e^{-(||p-c||-k)} & \text{otherwise.} \end{cases} \tag{10}$$

4.1.2 探针可见性。只有包围几何体的探针才会参与延迟着色。因此，距离几何表面较近的探针更为重要。同样，背向表面的纹素在着色时会被查询得更频繁。我们在公式11中将这两个量一并表示，其中 p 表示探针位置，$t = trace(p, -\omega)$。函数$trace$返回最近表面命中的距离，标量 s 则表示网格体素的对角线长度。

$$f_v(p, \omega) = e^{-2t/s} \tag{11}$$

4.1.3 入射辐射度。我们认为入射辐射度较高的方向更为重要。为识别这些方向，我们查询每个探针八分体上的辐射度，并将其作为入射辐射度的代表值。

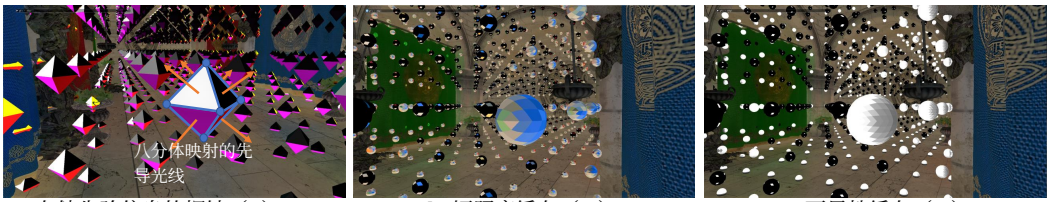
$$f_r(p, \omega) = \frac{min(r, \beta)}{\beta}, \tag{12}$$

其中 $r = lum(p, \omega)$。函数 亮度 使用命中表面点处的直接光照，返回入射亮度。参数 β 控制动态范围，我们设为 $\beta = 5$。

4.1.4 探针可见性变化。检测动态几何体对于在受这些变化影响的区域增加资源分配至关重要。我们通过计算探针可见性的时间梯度，并进行时空平滑操作来检测动态几何体。

$$f_0(p, \omega) = f_v^t(p, \omega) - f_v^{t-1}(p, \omega), \tag{13}$$

其中，f_v^t，f_v^{t-1}分别表示当前时间步和上一个时间步的可见性。公式 13 隐含地指出，在测量跨帧的时间差时，我们保持位置和方向不变，以避免噪声梯度。梯度经过时间触发器（Tr），如下所示：



a. 存储先验信息的探针（f）。 b. 辐照度缓存（$\hat{g}_r$）。 c. 可见性缓存（$\hat{g}_c$）。

图 5. 展示我们技术中各种探针映射纹理和查找表的图示。

$$f_1(p,\omega) = Tr\left(f_0(p,\omega),\theta\right), \tag{14}$$

其中，Tr将时间脉冲转换为由参数 θ控制的衰减信号，如图 6(a) 所示。为简洁起见，我们省略函数 Tr中的时间轴。该函数可最大限度地减少时间不连续性，从而帮助马尔可夫链在不同帧之间紧密跟随目标分布（h）。最后，我们进行如下空间卷积：

$$f_{\Delta v}(p,\omega) = \sum_{i,j} f_1(p - p_i, \omega - \omega_j). \tag{15}$$

卷积步骤可以平滑单个纹素中的不确定性，同时还可作为下一帧动态几何体可能位置的弱预测器。我们分别在空间和方向上使用 $5\times5\times5$ 和3×3 卷积。

4.1.5 探针辐亮度变化。 与上一节类似，我们使用探针辐亮度的时间梯度来检测辐射度变化。我们采用与上一节相同的时间触发器和空间卷积算子。相应方程如下：

$$f_2(p,\omega) = f_r^t(p,\omega) - f_r^{t-1}(p,\omega), \tag{16}$$

$$f_3(p,\omega) = Tr\left(f_2(p,\omega),\theta\right), \tag{17}$$

$$f_{\Delta r}(p,\omega) = \sum_{i,j} f_3(p - p_i, \omega - \omega_j). \tag{18}$$

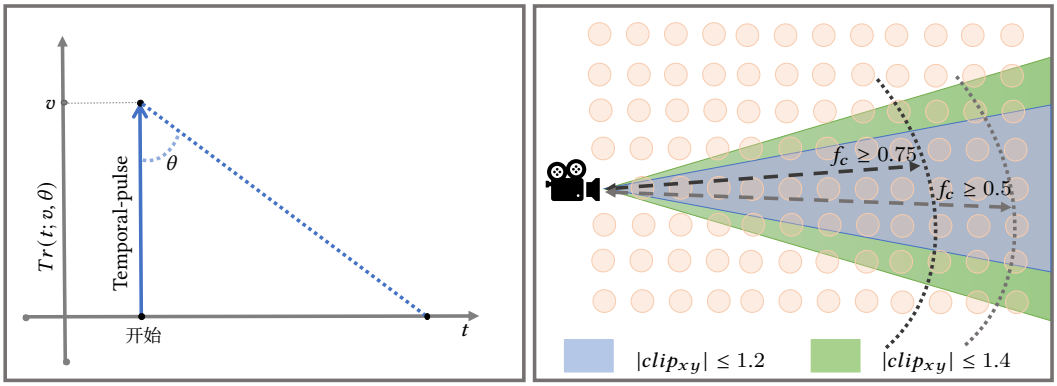
4.2 启发式组合

现在，单独的启发式方法已经定义完毕，如方程 1 所述，我们将其组合起来处理静态和动态情况，具体如下：

$$f_s(p,\omega) = \overbrace{f_c f_v}^{static}, \tag{19}$$

$$f_d(p,\omega) = \overbrace{f_c f_v (f_{\Delta v} + \mu f_{\Delta r})}^{dynamic}. \tag{20}$$

当环境保持静态时，我们根据相机以及探针到表面距离启发式进行采样，如方程 19 中的 f_c和 f_v所示。在方程 20 所表示的动态情况下，我们使用静态项 $f_c f_v$对环境中的变化进行调制。这种调制表明，我们更关注相机和几何表面附近发生的变化。因子 μ用于权衡几何体变化与光照变化的强度。在所有实验中，我们均使用 $\mu = 2$。



a. 将时间脉冲转换为衰减信号。　　b. 为探针定义裁剪体积。

图 6。图 (a) 展示了 temporal-trigger $Tr(v,\ \theta)$ 的构造。在图 (b) 中，我们将蓝色视锥与黑色边界所围成的体积称为内部体积 V_{in}。类似地，外部体积 V_{out} 是绿色视锥与外侧灰色边界所围成的体积。V_{out} 中的所有探针都参与启发式建模，如第 4.4 节所述。蓝色视锥内的探针参与自适应采样，如第 4.8、4.9 节所述。对于位于 V_{in} 之外但在 V_{out} 之内的所有探针，我们将其状态设为 $N = 16$，参见第 4.8 节。

4.3 启发式信息存储

我们将量 f_v，f_s，f_d 以 6-10-10 位编码的 32 位整数形式，存储在探针的每个八分体中。剩余的 6 位用于存储其他标志。查询查找表/纹理时，我们使用映射函数，将连续位置 p 和方向 ω 映射到查找表中对应的纹素。请注意，f_c 是隐式定义的，因此无需额外存储。

4.4 提高构建效率

如果对场景中的所有探针各追踪 8 条射线，启发式信息构建步骤可能会成为瓶颈。因此，如图 6(b) 所示，我们将先导光线限制在扩展相机视锥体内的探针上。为最大限度地提高算法效率，我们还会重复利用从 8 条先导光线中采集的采样，来填充辐照度（$\hat{g}_r$）和可见性（$\hat{g}_c$）缓存。我们按照 *AABBCCDD...* 模式，在交替的帧中改变光线方向，以提高对探针周围随时间变化的光场的检测能力。我们测量两帧之间的时间差（公式 13 和 16），这两帧具有相同的一组光线查询，从而避免噪声梯度。不过，这实际上会将检测频率减半（帧率 / *2*），但能提高空间感知能力。我们采用分层随机光线方向，确保每个八分体中始终有一条射线。我们在每隔一帧时更新辐照度和可见性缓存。

4.5 探针辐照度缓存

如图 5(b) 所示，辐照度缓存（$\hat{g}_r$）在空间中表示为均匀探针网格，其中每个探针使用球面映射，以 8×8 纹素分辨率存储周围的漫反射辐照度。对于每个纹素，我们使用自定义的 RGB 编码存储辐照度，三个色彩通道分别使用 9、9、8 位。剩余的 6 位（共 32 位）用于存储样本累积计数（N），以便计算样本流随时间变化的移动平均值（见算法 3）。在选择编码方式时，我们考虑了多个因素。编码应具备较高的带宽效率，并且必须支持在普通 GPU 上进行原子更新。我们发现，DX12 和 GLSL 都支持对 32 位整数执行原子操作。最后，我们的编码必须能够准确地

算法 3： 移动平均算法

输入： x：更新位置，v：新样本，N_{max}：最大样本数
输出： V：更新后的值，N：样本数

函数 MovingAvgUpdate(x, v, N_{max})**：**
- $n \leftarrow \hat{g}[x].N$ // 累计样本数
- $o \leftarrow \hat{g}[x].V$ // 累计值
- $V \leftarrow \frac{v}{n+1} + \frac{n \cdot o}{n+1}$ // 更新累计值
- $N \leftarrow min(n+1, N_{max})$ // 增加采样数
- **返回** V , N

编码超出标准定义范围的强度。我们对三个色彩通道应用非线性色彩压缩，$i \in [0..2]$ 如以下公式所示。

$$u_i = \frac{min\left(ln(\gamma \cdot v_i + 1), \beta\right)}{\beta}. \tag{21}$$

解码时，我们应用逆变换 $(exp(\beta \cdot u_i) - 1)/\gamma$，其中 $\beta = 5$和 $\gamma = 15$。有关我们选择压缩方案的更多详情，请参阅附录 B 和图11。

4.6 探针可见性缓存

如图5(c)所示，可见性探针中的纹素以 16x16 的纹素分辨率存储到最近几何体的平均距离和平均平方距离。我们将其称为 $\hat{g}_c$——即我们的可见性缓存。每个纹素用两个通道存储这两个值，每个通道的精度为 13 位，其余 6 位用于样本累积计数。我们使用探针笼对角线长度对距离进行归一化。与辐照度缓存类似，为了高效利用可用精度，我们按照公式 21 应用对数编码。对于线性通道和平方通道，我们分别使用 $(\beta,\ \gamma)$ 值 (5，15) 和 (8，20)。

4.7 时间样本累积机制

我们使用移动平均累积来存储辐照度缓存和可见性缓存中的采样。在算法 3 中，我们有两个参数 N和 N_{max}来控制移动平均累积。开始累积采样后，N会递增，算法的行为就像真正的移动平均。不过，当 N接近 $N_{max} - 1$时，算法会切换为带滞后的指数移动平均形式 ($N_{max} - 1)/N_{max}$。此外，请注意，当 N的值较低时，缓存更新较快，但存储的值可能会有噪声。随着 N增大，新样本对缓存的贡献权重会降低。我们利用这些参数控制静态和动态情况下的学习率与噪声，具体将在以下章节中讨论。

4.8 自适应采样 - 静态

我们将自适应采样策略分为两个阶段——静态和动态。我们使用两个独立的马尔可夫链集合，分别专注于捕捉周围光场的不同方面。静态链更注重准确性，而动态链则针对捕捉瞬态响应进行了调整。下一节将详细讨论动态链。

我们将方程 2 设置为 - $h = exp(min(\hat{g}_r/f_s,\ 1)) \cdot f_s$。来自辐照度缓存 $\hat{g}_r$的反馈取自上一帧以及更高的 Mip 级别（延迟着色器中也使用该级别）。最低的 Mip 级别 $\hat{g}_r$ 会持续更新，因此为避免在一帧内可能违反平稳性条件，我们不将其用作反馈。我们使用 Metropolis 采样算法 1 生成采样 $x_i \equiv (p_i,\ \omega_i)$。如算法 2 所述，我们使用这些采样来计算

算法 4： 原子移动平均算法

输入： x：更新位置，v：新的更新值
输出： 更新 $\hat{g}[x]$

```
函数 AtomicMovingAvg(x, v):
    当前 ← ĝ[x]
    /* 重复执行，直到目标值不再变化                                    */
    do
        预期值 ← 当前值
        下一个值 ← 移动平均更新(x, v, 64)
        互锁比较交换(ĝ[x], 预期值, 下一个值, 当前值)                    // 参考 HLSL
    而当前 ≠ 预期
```

连续光场 g_r，这需要沿方向 ω_i追踪一条从 p_i出发的射线。每个采样还会追踪一条额外的阴影射线，以计算相反方向的可见性（$-\omega_i$），因为延迟着色器中的探针对可见性的查询恰好与辐照度相位相差 $180°$。接下来，我们按照算法 4 所述，使用原子更新规则将辐照度和可见性值分别存入辐照度（$\hat{g}_r$）缓存和可见性（$\hat{g}_c$）缓存。由于链的多个调用可能会更新辐照度和可见性缓存中的同一位置，因此需要进行原子更新。图 4 总结了整体思路。

我们将随机游走步长（在算法 5 中记作 $\sigma \in R^5$）设置为与辐照度和可见性缓存中的离散化尺寸成正比。因此，位置步长与探针网格中体素的尺寸成正比，而角步长约为 $\sqrt{\pi}/256$。由于步长较小，缓存中的纹素可能会在每个纹素上累积多个采样，从而使样本数随时间大幅增加。我们还注意到，当样本数 $N = 0$ 达到 64 时，缓存的行为类似于真正的移动平均，这也有助于提高精度。

静态自适应样本有助于提高静态场景中的收敛速度，也适用于先验构建过程中未检测到的缓慢变化。例如游戏中的昼夜循环等缓慢的光照变化。我们通过为图 6(b) 中区域 $\{V_{out}\} - \{V_{in}\}$ 内的所有探针设置 $N = 16$，来降低滞后。这使探针能够快速追上最新值。

4.9 自适应采样 -动态

当检测到场景中存在动态内容时，我们会运行第二组马尔可夫链。存在动态元素，尤其是移动几何体时，会遇到两个主要问题。生成的样本在感兴趣区域内分布不均，也就是说，在存在时变变化的区域内分布不均。当步长较小时，链无法足够快地跟踪目标分布，因而无法从目标分布中生成样本，导致样本落后于不断移动的目标分布。第二个问题是对辐照度纹素进行多重采样所产生的噪声。理论上，可以通过增加滞后来改善时域样本复用，从而解决这一问题。然而，噪声降低的代价是引入令人难以接受的时域模糊。

我们通过增大链步长并粗化目标函数（f_d）来解决第一个问题。实际上，这相当于将启发式探针分组为虚拟代理。在我们的案例中，一个虚拟代理代表一组 $3 \times 3 \times 3$ 探针。这个虚拟探针有 8 个方向，每个方向代表一个轴对齐八分体。虚拟探针的一个纹素的值，是它沿相应方向所代表的全部 27 个探针中的最大值。此外，我们通过在公式 2 中设置 $\alpha = 0$ 来舍弃采样证据，因为过时的辐照度缓存（$\hat{g}_r$）对于

表 2. 展示我们技术所用各个场景的探针网格详情的表格。

场景	探针网格	探针间距 （单位：米）	辐照度（$\hat{g}_r$） 缓存分辨率	可见性（$\hat{g}_c$） 缓存分辨率
Bistro - Exterior	$192 \times 64 \times 192$	$0.5 \times 0.5 \times 0.5$	8×8	16×16
Sponza - Diffuse	$192 \times 64 \times 192$	$0.5 \times 0.5 \times 0.5$	8×8	16×16
Sponza - 光泽	$192 \times 64 \times 192$	$0.1 \times 0.1 \times 0.1$	16×16	16×16

表格 3. 此表展示了我们在比较中使用的各种技术的探针编码细节。

技术	辐照度 缓存编码	可见性 缓存编码	时间 滞后
Ours	$\lfloor R_9 \rfloor \lfloor G_9 \rfloor \lfloor B_8 \rfloor$ – N	$[R_{13}][G_{13}]$ – N	静态：0.98 ($N_{max} = 63$) 动态：0.91 ($N_{max} = 10$)
Q-DDGI	$\lfloor R_{11} \rfloor \lfloor G_{11} \rfloor \lfloor B_{10} \rfloor$ – N	$[R_{16}][G_{16}]$ – N	0.94
参考图像	RGB32f	RG32f	N/A

对随时间变化的区域进行采样几乎没有什么有用信息。与静态情况相比，链步长为 3 倍，而位置和方向的步长分别增大到 6 倍。

由于粗链中的每个采样代表整个八分体，因此我们会为组中所有底层的 3x3x3 探针，针对该八分体追踪 64 条光线。通过剔除未用于延迟着色的探针，我们提高了追踪步骤的效率。光线方向的调度是确定性的，会穿过辐照度缓存（$\hat{g}_r$）中某个纹素的中心。这样既解决了采样噪声问题，也有助于简化原子更新。由于这些光线并非随机光线，因此多个着色器调用同时更新同一个八分体并不会带来收益。因此，第一个更新该八分体的调用会以原子方式将其标记为已更新，使其他调用不会重复相同的工作，而是转到下一个八分体。

我们会在静态采样步骤之后运行动态采样。在静态采样期间，如果某个探针具有非零动态分量（$f_d > 0$），我们会将光线方向量化，使其穿过辐照度/可见性缓存纹素的中心，从而避免向纹素中注入采样噪声。

5 结果与比较

我们在不同场景中将结果与 Q-DDGI 和基于参考探针的实现进行比较，包括静态场景（图7）、动态几何体（图1、8、10）以及动态光照（图9）。

Q-DDGI：量化 DDGI（Q-DDGI）是对原始 DDGI[28], 的性能增强扩展，无需对基础算法进行重大修改即可实现。Q-DDGI 采用了更紧凑的辐照度和可见性缓存表示方式，与我们的方法十分相似。详见表3。我们还按照第4.4节和图6所述，在 Q-DDGI 中启用了探针的相机视锥体剔除。这些修改使 Q-DDGI 在不同场景中的性能（表4）与我们的方法相近，且探针数量相同（表2）。我们认为，这些修改使比较更加公平。我们为每个探针使用 32 条射线，每帧的总光线预算为 800–1600k 条射线（取决于场景）。

参考图像：参考实现使用标准的*FP32* 辐照度和可见性缓存表示，如表 3 所示。我们还使用分辨率更高的 32×32 辐照度和可见性缓存。受内存限制，我们只能使用尺寸为 $32 \times 32 \times 32$ 的较小探针网格，并采用与其他技术相同的探针间距（表 2）。每帧中，我们会丢弃探针中的所有先前值，并使用每纹素 64 条射线进行采样，以计算真实平均值。

我们的方法：我们使用 4096 个静态链调用实例和 1024 个动态链调用实例。总体而言，每帧使用 50 万至 90 万条射线（具体数量取决于场景）。

表4. 我们的方法与 Q-DDGI 的性能细分。我们的探针采样阶段分为三个子阶段——启发式构建（P）、静态自适应采样（S）和动态自适应采样（D）。

场景	我们的方法（毫秒）			Q-DDGI（毫秒）	
	探针采样（P +S +D）	延迟	探针采样总量	延迟	总计
Bistro - Exterior	4.01 + 2.23 + 4.73 = 11.0	4.63	15.6	22.3	4.47	26.8
Sponza - Diffuse	1.21 + 1.85 + 3.18 =6.24	3.62	9.86	9.69	3.51	13.2
Sponza - 光泽	4.83 + 2.11 + 4.33 =11.27	6.44	19.7	24.9	6.71	31.6

图1和图8展示了一个大型场景（Bistro Exterior），其中隧道的入口和出口经过修改，加入了动态闸门。隧道内壁仅由间接照明照亮，而间接照明受地面反射的直接光照控制。地面上的直接光照由动态入口闸门控制。该场景用于测试我们算法的跟踪能力；动态马尔可夫链应在移动门附近对探针进行采样。该场景还用于测试我们的颜色压缩方案在低光照和移动平均累积条件下的表现。

图9展示了动态光照下的 Sponza 场景，用于测试 ADGI 在没有动态几何体时的检测能力。图7展示了没有动态几何体或动态光照的静态场景，用于测试在未检测到动态变化或动态变化过于缓慢而无法检测时（例如游戏中的昼夜循环），我们的静态自适应采样的收敛情况。

图10展示了一个处于光泽间接光照下的动态几何体（斯坦福佛像），并以环境光照作为直接光照分量。该场景压力较大，因为与其他场景相比，相机视锥体中包含的探针数量多出许多倍，这是由于光泽光照需要更高的探针密度。此场景用于测试动态几何体在光泽地面上的瞬态响应。因此，该场景对时空模糊的容忍度较低。

我们在一台配备英伟达 2080Ti 图形处理器和 AMD 5600X 中央处理器的台式机上，以1920×1080 分辨率测量了结果。表4中引用的性能数据仅针对 ADGI 和 Q-DDGI 算法。G缓冲区和直接光照通道还分别需要额外的 2ms 和 3ms。

6 局限性

我们继承了原版 DDGI 算法的类似局限性。从着色点观察到的探针可见性只是近似值，需要通过探针移动等方式进行调整，以尽量减少光泄漏。探针表示无法高效捕捉光泽光传输；要捕捉光泽反射，需要对辐照度缓存进行密集的空间角度离散化。

准确检测场景中的瞬态时空变化十分困难。对于动态几何体，动态物体距离探针越远，检测精度就越低。动态光照也是如此；尤其是距离探针较远的高频局部照明，很难检测。此外，要让马尔可夫链跟踪目标分布，运动速度应限制在与马尔可夫链步长和平均帧率的乘积相当的范围内。尽管许多游戏引擎会跟踪动态物体，从而便于检测可见性的变化，但我们仍需要使用光线追踪来检测动态辐射度。

7 结论

我们的自适应采样方法提高了原始 DDGI 算法的效率。该方法会在存在时变现象的区域非均匀地分配资源，并能捕捉包含数百万个探针的环境中的瞬态局部变化。相比之下，DDGI 的均匀分配策略会分散关键区域的资源，探针数量较多时尤其如此。这些改进减少了时间滞后，也降低了依靠时域模糊来减少噪声的需求。我们的探针编码方案将内存需求降低了 4 倍（相应地也降低了内存带宽需求），对画质的影响极小，同时还支持在场景中使用数百万个探针。我们的自适应采样阶段的计算需求设有固定上限，并且使采样与探针数量脱钩，进一步降低了内存带宽需求。这些变化提升了基于探针的渲染效果，同时带来了 1.5–2 倍的性能提升。

8 相关工作扩展

辐照度缓存 辐照度缓存是另一类旨在克服全局光照高计算成本的技术。辐照度缓存方法假设辐照度在场景中平滑变化，并可通过反照率调制恢复纹理细节 [64]。各种缓存记录的插值与位置至关重要，尤其是在平滑性假设不成立时。虽然已有稳健且有理论依据的离线解决方案 [16, 24], , 但实时应用往往诉诸复杂的启发式方法，并施加严格限制以实现在线全局光照。压缩 [56], 、稀疏插值 [49], 、预卷积环境贴图 [42, 45], 、空间哈希 [3] 以及使用神经网络 [37] ，都是实时辐照度缓存技术的进展实例。尽管这些方法旨在实现实时性能，但其复杂性和限制使得实现与部署颇具挑战。

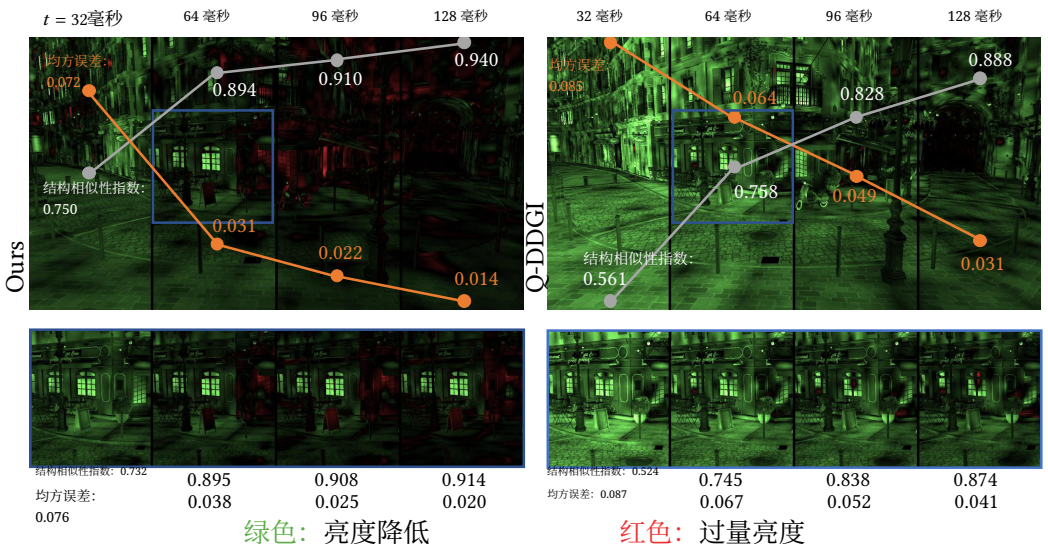


图7. 对比我们的技术在静态Bistro Exterior场景中随时间的收敛情况。该图展示了我们的静态自适应采样步骤的有效性。两行图像测量了相对于参考图像的亮度差异，并以红色和绿色突出显示误差。

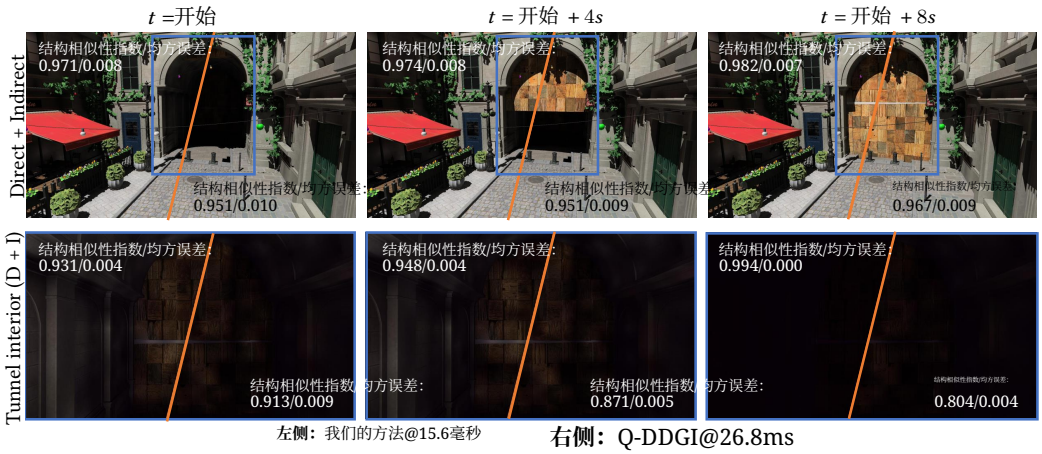


图 8. 在添加了移动门的修改版 Bistro Exterior 场景中，将我们的方法与 Q-DDGI 进行比较。该场景包含 $192 \times 64 \times 192$ 个探针，展示了两种技术在场景动态区域附近的收敛情况。第二行展示了随着门逐渐关闭，隧道内部发生的变化。我们的方法能够更好地将资源分配到更靠近动态区域的位置，从而实现更快的收敛速度和更高的性能。

路径追踪 路径追踪所提供的灵活性和通用性 [18] 对于实时渲染极具吸引力。然而，由于计算需求庞大，路径追踪一直难以应用于实时场景。即使有了硬件加速光线追踪 [23], , 实时情况下每个像素也只能追踪几十条光线。因此，高效的采样策略和高质量的降噪算法 [38, 46, 47] 至关重要。许多采样方法试图在渲染过程中学习入射光照的表示 [1, 8, 34, 44, 60]。虽然这些方法能够显著降低误差，但在图形处理器上并行构建这些结构会产生大量开销，似乎不适用于实时应用。近期提出的 ReSTIR GI [41] 通过在空间和时间上重用路径，提供了一种高效的实时采样策略；但经过第二次反弹后，算法会变得复杂，且最终阶段仍需要降噪。深度学习也已应用于路径引导，包括 [35, 36]的相关研究。这些方法通过更有效的路径采样显著降低了误差，但其性能仍不足以满足实时应用的需求。

屏幕空间方法： 在游戏中，利用屏幕空间方法以实时帧率近似实现符合物理规律的光照颇受欢迎。屏幕空间方法速度快、对图形处理器友好，而且易于实现。屏幕空间环境光遮蔽（SSAO） [2, 33] 已应用于许多实时渲染引擎。在 SSAO 的基础上，屏幕空间方向性遮蔽（SSDO） [43] 用于近场的直接和间接漫反射光照。Sousa 等人 [52] 提出了屏幕空间反射（SSR），通过直接在屏幕空间中采用二维光线追踪方法来获取间接镜面分量。近年来，屏幕空间全局光照（SSGI） [43, 50, 52] 方法为实时全局光照提供了可行的解决方案。然而，这些方法受限于观察者位置可见的信息，因此难以设计出稳健的解决方案。

重要性采样与贝叶斯建模： 重要性采样根据先验知识（即有依据的猜测），有选择地评估被积函数中的元素，从而提供了一种降低暴力积分成本的方法。以往关于重要性采样的研究提出了不同方法，将重要性采样应用于渲染方程 [21, 48, 57]中的各种蒙特卡罗积分。尽管马尔可夫链蒙特卡罗（MCMC）方法

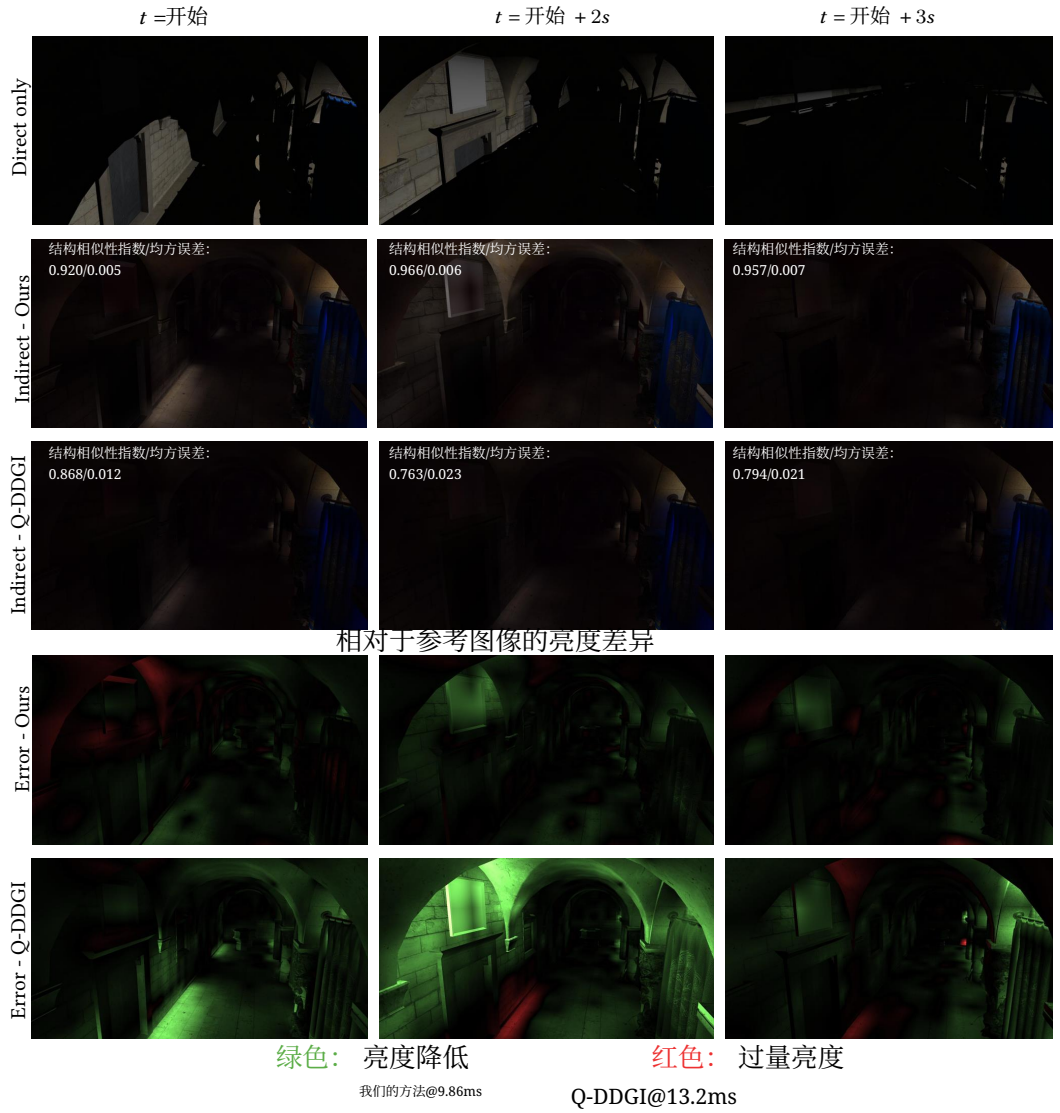


图 9. 对比动态光照下我们的方法收敛情况的图示，其中动态光照由第一行所示的直接分量控制。最后两行测量相对于参考图像的亮度差异，并以红色和绿色突出显示误差。

早在神经网络发展的初期就已用于贝叶斯学习 [39], , 而随机梯度马尔可夫链蒙特卡罗也已被提出 [65] , 并应用于各种领域 [25], , 但我们的方法既不基于蒙特卡罗，也不依赖神经网络学习。我们通过定义引导函数（先验）、似然和后验，利用贝叶斯推断与马尔可夫链作为数学工具，对探针上的重要纹素进行采样。

马尔可夫链： 马尔可夫链广泛应用于蒙特卡罗路径追踪。例如，Veach 和 Guibas [58] 使用 Metropolis 采样来探索所有可能路径构成的空间。Kelemen 等人 [19] 随后在随机数空间中应用了精确采样，即在主样本

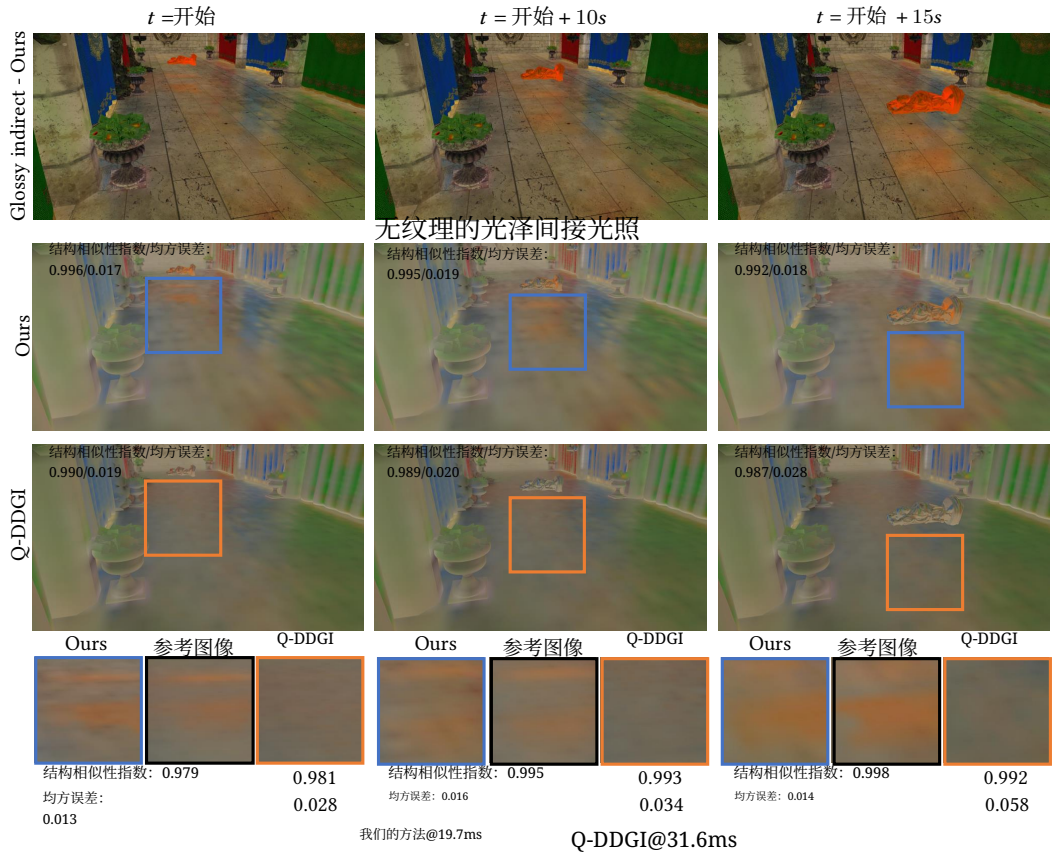


图 10. 比较光泽间接反射的示意图。场景由环境光照明。该场景测试了移动的佛像几何体在光泽地面上产生的瞬态响应。

空间中。Bitterli 等人最新的研究 [4] 将简单的路径追踪积分器与马尔可夫链蒙特卡罗相结合，使用高方差路径的随机种子作为马尔可夫链的起始点。尽管马尔可夫链在解决蒙特卡罗采样问题时发挥了显著作用，但我们对采样以及利用马尔可夫链从引导函数中抽取样本的看法与之不同。

贝叶斯推断：贝叶斯建模是计算机视觉和图形学中广泛应用的方法。Brouillat 等人 [5] 和 Marques 等人 [30] 率先将贝叶斯蒙特卡洛（BMC） [11] 应用于光传输模拟。相比之下， [59] 保留了高效的经典频率学派蒙特卡罗方法，并运用贝叶斯建模来优化采样分布，从而估计场景中的直接光照。Vorba 等人 [61], 也采用了类似的方法，他们使用最大后验（MAP）公式，对参数混合模型的训练进行正则化，以优化间接光照采样。我们的方法在光照探针领域运用贝叶斯建模，基于采样证据检测重要的探针和方向。

参考文献

[1] 2019. SIGGRAPH '19：ACM SIGGRAPH 2019 制作会议（洛杉矶，加利福尼亚州）。计算机协会，美国纽约州纽约。

[2] Louis Bavoil、Miguel Sainz 和 Rouslan Dimitrov。2008。图像空间基于视界的环境光遮蔽。载于 ACM SIGGRAPH 2008 Talks（洛杉矶，加利福尼亚州）（SIGGRAPH '08）。计算机协会，美国纽约州纽约，文章 22, 1 10 1145 1401032 1401061 页。https://doi.org/ 。/ 。Nikolaus Binder、Sascha Fricke 和 Alexander Keller。 。通过抖动空间哈希进行快速路径空间滤波。载于 ACM SIGGRAPH 2018 Talks（温哥华，不列颠哥伦比亚省，加拿大）（SIGGRAPH '18）。计算机协会，美国纽约州纽约，文章 71, 2 10 1145 3214745 3214806 页。https://doi.org/ 。/ 。Benedikt Bitterli 和 Wojciech Jarosz。 。选择性Metropolis蒙特卡洛光传输模拟。《ACM图形学汇刊》（TOG）38, 6（2019），1–10。[5] Jonathan Brouillat、Christian Bouville、Brad Loos、Charles Hansen 和 Kadi Bouatouch。2009。一种用于全局光照的贝叶斯蒙特卡洛方法。载于《计算机图形学论坛》，第28卷。Wiley Online Library，美国，2315–2329。[6] Chakravarty R. Alla Chaitanya、Anton S. Kaplanyan、Christoph Schied、Marco Salvi、Aaron Lefohn、Derek Nowrouzezahrai 和 Timo Aila。2017。使用循环去噪自编码器对蒙特卡洛图像序列进行交互式重建。《ACM图形学汇刊》36, 4，文章98（2017年7月），12页。https://doi.org/10.1145/ 3072959.3073601[7] Siddhartha Chib 和 Edward Greenberg。1995。理解 Metropolis-Hastings 算法。《美国统计学家》49, 4（1995），327–335。[8] Stavros Diolatzis、Adrien Gruson、Wenzel Jakob、Derek Nowrouzezahrai 和 George Drettakis。2020。使用线性变换余弦的实用乘积路径引导。载于《计算机图形学论坛》（ Eurographics 渲染研讨会论文集）39, 4（2020年7月）。[9] William Donnelly 和 Andrew Lauritzen。2006。方差阴影贴图。载于《2006年交互式三维图形与游戏研讨会论文集》（加利福尼亚州雷德伍德城）（I3D '06）。计算机协会，美国纽约州纽约，161 165 10 1145 1111411 1111440 – 。https://doi.org/ 。/ 。Charles J Geyer。 。实用马尔可夫链蒙特卡洛。，–483页。[11] Zoubin Ghahramani 和 Carl Rasmussen。2002。贝叶斯蒙特卡洛。https://proceedings.neurips.cc/paper/2002/ file/24917db15c4e37e421866448c9ab23d8-Paper.pdf[12] Toshiya Hachisuka、Wojciech Jarosz、Richard Peter Weistroffer、Kevin Dale、Greg Humphreys、Matthias Zwicker 和 Henrik Wann Jensen。2008。光线追踪的多维自适应采样与重建。《ACM图形学汇刊》27, 3 2008 1 10 10 1145 1360612 1360632 （8月），– 。https://doi.org/ 。/ 。Jon Hasselgren、J. Munkberg、Marco Salvi、A. Patney 和 Aaron Lefohn。2020。神经时域自适应采样与去噪。《计算机图形学论坛》39 05 2020 147 155 10 1111 13919 （），– 。https://doi.org/10.1111/cgf. Julius Ikkala、Petrus Kivi、Joel Alanko、Markku Mäkitalo 和 Pekka Jääskeläinen。 。DDISH-GI：动态分布式球谐全局光照。载于《计算机图形学进展：第38届国际计算机图形学会议（CGI 2021），线上会议，2021年9月6日至10日，论文集》。Springer-Verlag，德国柏林海德堡，433–451。https://doi.org/ 10 1007 978 3 030 89029 2_34 . / - - - - Michał Iwanicki 和 Peter-Pike Sloan。 。《使命召唤：无限战争》中的预计算光照。[16] Wojciech Jarosz、Craig Donner、Matthias Zwicker 和 Henrik Wann Jensen。2008。参与介质的辐亮度缓存。《ACM图形学汇刊》27, 1, 7 2008 11 10 1145 1330511 1330518 文章（3月），页。https://doi.org/ 。/ 。Henrik Wann Jensen。 。使用光子图实现全局光照。 ，–页。[18] James T. Kajiya。1986。渲染方程。载于《第13届年度计算机图形学与交互技术会议论文集》（SIGGRAPH '86）。计算机协会，美国纽约州纽约，143–150。https://doi.org/ 10 1145 15922 15902 . / 。Csaba Kelemen、László Szirmay-Kalos、György Antal 和 Ferenc Csonka。2002。适用于 Metropolis光传输算法的简单稳健变异策略。 ，531–540页。[20] Alexander Keller。1997。即时辐射度。载于《第24届年度计算机图形学与交互技术会议论文集》（SIGGRAPH '97）。ACM Press/Addison-Wesley Publishing Co.，美国，49–56。https://doi.org/10. 1145/258734.258769[21] A. Keller、L. Fascione、M. Fajardo、I. Georgiev、P. Christensen、J. Hanika、C. Eisenacher 和 G. Nichols。2015。电影行业中的路径追踪革命。载于 ACM SIGGRAPH 2015 Courses（洛杉矶，加利福尼亚州）（SIGGRAPH '15）。计算机协会，美国纽约州纽约，文章 24, 7 10 1145 2776880 2792699 页。https://doi.org/ 。/ 。Alexander Keller、Timo Viitanen、Colin Barré-Brisebois、Christoph Schied 和 Morgan McGuire。 。我们已经完成光线追踪了吗？载于 ACM SIGGRAPH 2019 Courses（洛杉矶，加利福尼亚州）（SIGGRAPH '19）。计算机协会，美国纽约州纽约，文章 3, 381 10 1145 3305366 3329896 页。https://doi.org/ 。/ 。[23] Alexander Keller 和 Carsten Waechter。 。实时精确光线追踪。https://patents.google.com/patent/US20070024615A1/en 美国专利US20070024615A1。[24] Jaroslav Křivánek、Pascal Gautron、Greg Ward、Henrik Wann Jensen、Per H. Christensen 和 Eric Tabellion。2008。使用辐照度缓存的实用全局光照。载于 ACM SIGGRAPH 2008 Classes（洛杉矶，加利福尼亚州）（SIGGRAPH '08）。计算机协会，美国纽约州纽约，文章60，20页。https://doi.org/10.

1145/1401132.1401213[25] Chunyuan Li、Andrew Stevens、Changyou Chen、Yunchen Pu、Zhe Gan 和 Lawrence Carin。2016。使用随机梯度马尔可夫链蒙特卡罗进行形状分类的权重不确定性学习。第5666–5675页。https://doi.org/10.1109/CVPR.2016.611[26] Daqi Lin、Chris Wyman 和 Cem Yuksel。2021。使用时空储层重采样进行快速体渲染。ACM图形学汇刊 40, 6, 279 2021 18 10 1145 3478513 3480499 文章（dec），页。[27] https://doi.org/ . / . Daqi Lin 和 Cem Yuksel。. 实时随机光照切割。ACM计算机图形学与交互技术论文集3，1，文章 5 2020 18 10 1145 3384543 （apr），页。https://doi.org/ . / Zander Majercik、Jean-Philippe Guertin、Derek Nowrouzezahrai 和 Morgan McGuire。2019。利用光线追踪辐照度场实现动态漫反射全局光照。计算机图形技术期刊（JCGT）8，2（2019年6月5日），1–30。http://jcgt.org/published/0008/02/01/ [29] Zander Majercik、Adam Marrs、Josef Spjut 和 Morgan McGuire。2021。面向生产环境扩展基于探针的实时动态全局光照。计算机图形技术期刊（JCGT）10，2（2021年5月3日），1–29。http://jcgt.org/published/0010/02/01/ [30] Ricardo Marques、Christian Bouville、Mickaël Ribardière、Luís Paulo Santos 和 Kadi Bouatouch。2013。用于光泽表面贝叶斯蒙特卡罗渲染的球面高斯框架。IEEE可视化与计算机图形学汇刊 19, 10 2013 1619 1632 10 1109 2013 79 （），–。[31] https://doi.org/ . /TVCG. . Morgan McGuire、Mike Mara、Derek Nowrouzezahrai 和 David Luebke。. 使用预计算光场探针实现实时全局光照。载于第21届ACM SIGGRAPH交互式三维图形与游戏研讨会论文集（加利福尼亚州旧金山）（I3 2，11 17 10 1145 3023378 3023368D ’）。计算机协会，纽约州纽约，美国，文章页码。https://doi.org/ . / .[32] Soham Uday Mehta、Brandon Wang 和 Ravi Ramamoorthi。2012。用于交互式采样软阴影的轴对齐滤波。ACM图形学汇刊 31, 6, 163 2012 10 10 1145 2366145 2366182 文章（Nov.），页。[33] https://doi.org/ . / . Martin Mittring。. 探索次世代：CryEngine。载于ACM SIGGRAPH课程（加利福尼亚州圣迭戈）（SIGGRAPH ’07）。计算机协会，纽约州纽约，美国，97–121。https://doi.org/10.1145/1281500. 1281671[34] Thomas Müller、Markus Gross 和 Jan Novák。2017。用于高效光传输模拟的实用路径引导。计算机图形学论坛（EGSR论文集） 36, 4 2017 91 100 10 1111 13227（June），–。[35] https://doi.org/ . /cgf. Thomas Müller、Brian Mcwilliams、Fabrice Rousselle、Markus Gross 和 Jan Novák。. 神经重要性采样。ACM图形学汇刊 38, 5, 145 2019 19 10 1145 3341156 文章（oct），页。[36] https://doi.org/ . / Thomas Müller、Fabrice Rousselle、Jan Novák 和 Alexander Keller。. 用于路径追踪的实时神经辐射缓存。ACM图形学汇刊 40, 4, 36 2021 16 10 1145 3450626 3459812 文章（jul），页。[37] https://doi.org/ . / . Thomas Müller、Fabrice Rousselle、Jan Novák 和 Alexander Keller。. 用于路径追踪的实时神经辐射缓存。ACM图形学汇刊 40, 4, 36 2021 16 10 1145 3450626 3459812 文章（jul），页。[38] https://doi.org/ . / . Jacob Munkberg 和 Jon Hasselgren。. 使用层嵌入进行神经去噪。载于《计算机图形学论坛》，第39卷。Wiley Online Library, 1–12。[39] Radford M Neal。2012。神经网络的贝叶斯学习。第118卷。施普林格科学与商业媒体，美国。[40] Y. Ouyang、S. Liu、M. Kettunen、M. Pharr 和 J. Pantaleoni。2021。ReSTIR GI：面向实时路径追踪的路径重采样。计算机图形学论坛40，8（2021），17–29。https://doi.org/10.1111/cgf.14378 arXiv:https://onlinelibrary.wiley.com/doi/pdf/ 10 1111 14378 . /cgf. Yaobin Ouyang、Shiqiu Liu、Markus Kettunen、Matt Pharr 和 Jacopo Pantaleoni。2021。ReSTIR GI：面向实时路径追踪的路径重采样。载于《计算机图形学论坛》，第40卷。Wiley Online Library，17–29。[42] Hauke Rehfeld、Tobias Zirr 和 Carsten Dachsbacher。2014。聚类预卷积辐射缓存。载于第14届Eurographics并行图形与可视化研讨会论文集（威尔士斯旺西，英国）（PGV ’14）。Eurographics协会，德国戈斯拉尔，25–32。[43] Tobias Ritschel、Thorsten Grosch 和 Hans-Peter Seidel。2009。在图像空间近似动态全局光照。载于2009年交互式三维图形与游戏研讨会论文集（马萨诸塞州波士顿）（I3 09 75 82 10 1145 1507149 1507161 D ’）。计算机协会，纽约州纽约，美国，–。https://doi.org/ . / . Lukas Ruppert、Sebastian Herholz 和 Hendrik P. A. Lensch。. 用于路径引导的视差感知混合分布稳健拟合。ACM图形学汇刊 39, 4, 147 2020 15 10 1145 3386569 3392421 文章（jul），页。[45] https://doi.org/ . / . Daniel Scherzer、Chuong Nguyen、Tobias Ritschel 和 Hans-Peter Seidel。. 预卷积辐射缓存。计算机图形学论坛（2012）。https://doi.org/10.1111/j.1467-8659.2012.03134.x [46] Christoph Schied、Anton Kaplanyan、Chris Wyman、Anjul Patney、Chakravarty R. Alla Chaitanya、John Burgess、Shiqiu Liu、Carsten Dachsbacher、Aaron Lefohn 和 Marco Salvi。2017。时空方差引导滤波：路径追踪全局光照的实时重建。载于高性能图形学论文集（加利福尼亚州洛杉矶）（HPG ’17）。计算机协会，纽约州纽约，美国，文章2，12页。https://doi.org/10.1145/3105762.3105770

[47] Christoph Schied、Christoph Peters 和 Carsten Dachsbacher。2018。用于实时自适应时域滤波的梯度估计。ACM计算机图形学与交互技术论文集1，2（2018），1–16。[48] Pradeep Sen 和 Soheil Darabi。2012。关于滤除蒙特卡罗渲染中随机参数噪声的研究。ACM图形学汇刊31，3（2012），18–1。[49] Ari Silvennoinen 和 Jaakko Lehtinen。2017。利用稀疏辐射度探针进行预计算局部重建，实现实时全局光照。ACM图形学汇刊36，6，文章230（nov 2017），13页。https://doi.org/10.1145/ 3130800.3130852[50] Ari Silvennoinen 和 Ville Timonen。2015。《量子破碎》中的多尺度全局光照。[51] Peter-Pike Sloan、Jan Kautz 和 John Snyder。2002。针对动态低频光照环境的预计算辐射传输，实现实时渲染。载于第29届年度计算机图形学与交互技术会议论文集（德克萨斯州圣安东尼奥）（SIGGRAPH '02）。计算机协会，纽约州纽约，美国，

527 536 10 1145 566570 566612 –。https://doi.org/ . / . Tiago Sousa、Nick [52] 2011
Kasyan 和 Nicolas Schulz。. CryENGINE 3图形技术揭秘。[53] Michael Stengel、Zander Majercik、Benjamin Boudaoud 和 Morgan McGuire。2021。用于向瘦客户端无损流式传输动态光照探针的分布式解耦系统。载于第12届ACM多媒体系统会议论文集（土耳其伊斯坦布尔）（MMSys '21）。计算机协会，纽约州纽约，美国，159–172。https://doi.org/ 10 1145 3458305 3463379 . / . Eric Tabellion 和 Arnauld Lamorlette。2004。面向计 [54]
算机生成影片的近似全局光照系统。载于ACM SIGGRAPH 2004论文集（加利福尼亚州洛杉矶）（SIGGRAPH '04）。计算机协会，纽约州纽约，美国， 469 476 10 1145 1186562 1015748 –。
https://doi.org/ . / . Natalya Tatarchuk。. 面向游戏的辐照度体，页。[56] Kostas Vardis、Georgios Papaioannou [55] 2005 0
和 Anastasios Gkaravelis。2014。使用色度压缩的实时辐射度缓存。计算机图形技术期刊（JCGT）3，4（2014年12月16日），111–131。http://jcgt.org/ published/0003/04/06/ [57] Eric Veach 和 Leonidas J. Guibas。1995。蒙特卡罗渲染中采样技术的最优组合。载于第22届年度计算机图形学与交互技术会议论文集（SIGGRAPH '95）。计算机协会，纽约州纽约，美国，

419 428 10 1145 218380 218498 –。
[58] 1997 24
https://doi.org/ . / . Eric Veach 和 Leonidas J. Guibas。 . 梅特ropolis光传输。载于第届年度计算机图形学与交互技术会议论文集（SIGGRAPH '97）。ACM出版社/艾迪生-韦斯利出版公司，美国，

65 76 10 1145 258734 258775 –。https://doi.org/ . / . Petr Vévoda、Ivo Kondapaneni 和 Jaroslav [59]
Křivánek。2018。用于自适应直接光照采样的贝叶斯在线回归。ACM图形学汇刊（TOG）37，4（2018），1–12。[60] Jiří Vorba、Ondřej Karlík、Martin Šik、Tobias Ritschel 和 Jaroslav Křivánek。2014。用于光传输模拟的参数混合模型在线学习。ACM图形学汇刊33，4，文章101（jul 2014），11页。https: //doi.org/
10 1145 2601097 2601203 . / . Jiří Vorba、Ondřej Karlík、Martin Šik、Tobias Ritschel 和 Jaroslav [61]
Křivánek。2014。用于光传输模拟的参数混合模型在线学习。ACM图形学汇刊33，4，文章101（jul 2014），11页。https: //doi.org/ 10 1145 2601097 2601203 . / . Bruce Walter、Sebastian Fernandez、Adam Arbree、 [62]
Kavita Bala、Michael Donikian 和 Donald P. Greenberg。2005。Lightcuts：一种可扩展的光照方法。ACM图形学汇刊24，3（jul 2005），1098–1107。https://doi.org/10. 1145/1073204.1073318[63] Yue Wang、Soufiane Khiat、Paul G. Kry 和 Derek Nowrouzezahrai。2019。快速非均匀辐射度探针布置与轨迹追踪。载于ACM SIGGRAPH交互式三维图形与游戏研讨会论文集（加拿大魁北克省蒙特利尔）（I3D '19）。计算机协会，纽约州纽约，美国，文章5，9页。https: //doi.org/ 10 1145 3306131 3317024 . / . Gregory J. Ward、Francis M. Rubinstein 和 Robert [64]
D. Clear。1988。漫反射互反射的光线追踪解法。SIGGRAPH计算机图形学
22, 4 1988 85 92 10 1145 378456 378490 （jun ），–。
[65] 2011
https://doi.org/ . / . Max Welling 和 Yee Whye Teh。. 基于随机梯度朗之万动力学的贝叶斯学习。载于第28届国际机器学习会议论文集（美国华盛顿州贝尔维尤）（ICML'11）。Omnipress，美国威斯康星州麦迪逊，681–688。[66] Lei Xiao、Salah Nouri、Matt Chapman、Alexander Fix、Douglas Lanman 和 Anton Kaplanyan。2020。用于实时渲染的神经超采样。ACM图形学汇刊39，4，文章142（2020年7月），12页。https: //doi.org/10.1145/3386569.3392376

A METROPOLIS-HASTINGS

马尔可夫链蒙特卡洛（MCMC）无需计算边缘分布即可从后验中进行采样。 [10]。Metropolis-Hastings（Metropolis）是我们在本研究中采用的一种 MCMC 具体实现 [7]。只要给定一个与概率密度 $P(x)$ 成正比的函数 $h(x)$，Metropolis-Hastings 算法就能从任意概率分布的概率密度 $P(x)$ 中抽取样本。Metropolis 算法通过生成一系列样本值来运行；随着样本数量增加，样本的分布会越来越接近所需的分布。这些样本值是迭代生成的，也就是说，下一个样本取决于当前样本（因此样本序列构成一条链）。设 $h(x)$ 是一个与所需概率密度函数 $P(x)$（亦称目标分布）成正比的函数。带随机游走的 Metropolis 马尔可夫链算法定义如下：

算法 5： 随机游走算法

输入： x^i：当前状态，y^i：当前状态的概率
输入： σ：步长或高斯噪声的标准差
输出： x^{i+1}：下一个状态，y^{i+1}：下一个状态的概率

函数 随机游走(x^i, y^i)**：**
- $x^{i+1} \leftarrow x^i + \mathrm{N}(\sigma)$　// 提议一个新状态
- $y^{i+1} \leftarrow h(x^{i+1})$
- $\mu \leftarrow$ 最小值$\left\{\frac{y^{i+1}}{y^i}, 1\right\}$　// 计算接受率
- $\epsilon \sim U(0,1)$　// 对均匀分布进行采样
- **若** $\epsilon > \mu$ **则**
  - /* 拒绝提议的状态　*/
  - $x^{i+1} \leftarrow x^i$
  - $y^{i+1} \leftarrow y^i$
- **返回** x^{i+1}，y^{i+1}

初始化： 在样本空间中任意选择一个点 x^{i-1} 作为初始观测，并选择一个任意的概率密度 $\mathrm{N}(x^i \mid x^{i-1})$，根据前一个样本值 x^{i-1}提出下一个候选样本 x^i。在我们的工作中，假设 N 是对称的。一种常见的选择是令 $\mathrm{N}(x^i \mid x^{i-1})$ 为以 x^{i-1} 为中心的高斯分布，这样，距离 x^{i-1} 更近的点更有可能被访问，从而使样本序列类似于随机游走 [7]。随机游走算法如算法 5 所述。

B 探针压缩

我们测试了几种 26 位编码，最终选定了图11中以⌊R9⌋⌊G9⌋⌊B8⌋ – N表示的非线性 RGB 编码。在这种编码中，RGB 颜色首先按照公式21经过对数非线性变换，使量化误差均匀分布在各个强度范围内。我们对所有通道都采用向下取整 (⌊⌋)量化，尽管四舍五入 ([ ])更加精确。我们的量化方案可确保移动平均更新在新样本的强度较低时生成暗色。在四舍五入 设置下，由于向上舍入 误差，颜色可能永远无法变为零。有趣的是，YCbCr 编码允许对 Y 通道向下舍入，并对 Cb 和 Cr 通道四舍五入；然而，如图11所示，这种编码在亮度和颜色保真度指标方面的表现都很差。

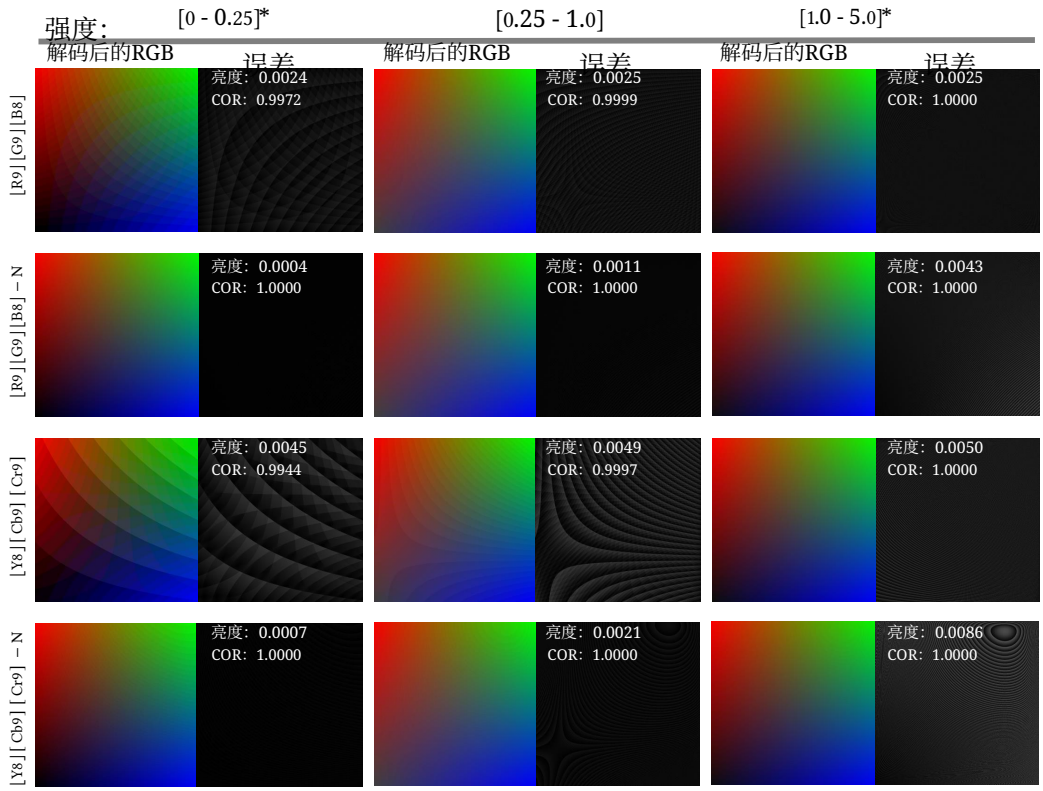


图11。比较具有动态范围的三维色彩空间切片上26位颜色编码的图示。我们将以RGB32f参考为基准测得的亮度和颜色相关性重建误差进行比较。标记有-N后缀的对数非线性编码将位误差从较低强度转移到较高强度，而间接照明中较高强度出现的频率较低。 ⌊⌋ 和 [ ] 分别表示向下取整量化和就近取整量化。* 色彩图可视化已进行归一化处理。

公式21中的参数通过网格搜索获得：针对各种颜色和强度组合，最小化相对于RGB32f参考的重建误差，如图11所示。亮度误差是两张色彩图之差的均方根值。颜色准确度则通过两张展平色彩图之间的归一化点积来衡量。