

一种可扩展的和生产就绪的天空和大气渲染技术

塞巴斯蒂安希拉尔¹

¹ 史诗游戏公司



图1: 使用本文提出的方法渲染不同大气条件和视图的图像。从左到右: 类似地球的白天和类似火星的蓝色日落, 类似地球的太空景观和一个小行星的艺术视野。

摘要

我们提出了一种基于物理的方法来渲染平面的大气从地面到空间视图。我们的方法计算起来很便宜, 而且与以前成功的方法相比, 不需要任何高维查找表 (lut), 因此不需要与之相关的视觉伪影。我们还提出了一种新的近似方法来实时评估光在大气中的多重散射。我们重新审视渲染自然大气效果的含义, 并提出一套简单的查找表和参数化来渲染天空和它的空中视角。大气成分可以动态变化, 以匹配艺术视觉和天气变化, 而不需要大量的LUT更新。完整的技术可以用于实时应用程序, 如游戏、模拟器或架构预可视化。这个该技术还可以从节能的移动平台扩展到具有高端gpu的pc, 而且也可以用于加速路径跟踪。

中国化学会概念
Subarg计算方法! 射线跟踪

1. 介绍

渲染自然现象对于可信世界的视觉模拟是很重要的。大气模拟和渲染对于需要具有动态时间的大型开放世界或从太空观察行星的应用程序是很重要的。这些应用程序包括游戏、建筑可视化和飞行或空间模拟器。然而, 目前的方法有局限性: 它们要么局限于从地面观看, 只能代表单一的大气类型, 在大气属性改变时需要昂贵的查找表 (LUTs) 更新, 甚至可以显示视觉伪影。

我们提出了一种方法来呈现一个行星的天空和空中的有机玻璃-

从大气的参与媒介的实时表示。我们在本文中的贡献如下:

- 我们提出了一种天空和空中透视渲染技术
躺在灯具上, 以较低的分辨率评估照明整体的昂贵部分, 同时保持重要的视觉特征。
- 我们提出了一种新的方法来评估大气中光的多重散射的贡献。
它可以近似出无限个散射阶数, 也可以用于加速路径跟踪。
- 该技术支持一天中的动态时间以及大气属性的动态更新, 同时渲染效率-

这是在一个广泛的设备上，从低端的苹果iPhone 6s到游戏机和高端的游戏个人电脑。

这种方法被用于史诗游戏的虚幻引擎中。在本文中，我们将使用光度测量单位（亮度/照度），而不是辐射测量单位（辐照度/辐照度）。这是由于这些术语在现代游戏引擎中的普遍存在。

在回顾了第2节中之前的工作之后，我们在第3节中简要描述了参与的媒体渲染（重点关注大气的情况）。本文中使用的材料模型见第4节，我们的大气渲染技术详见第5节。第6节将讨论结果和与路径追踪的地面真相和以前的模型的比较。最后，我们在第7节中报告了绩效，并得出结论。

2. 以前的工作

第一波天空渲染技术集中于从这个角度行进大气层的射线。这就是西田等人的观点。[NSTN93]最初提出作为一种从地面和空间视图呈现大气的方法。O'Neil [ONe07]为了提高性能，建议对每个顶点的散射亮度进行积分，并使用每个像素的相位函数来渲染最终的天空颜色。Wenzel [Wen07]提出了同样的想法，但散射亮度存储在纹理中，在几帧上更新以摊销成本。这些模型的主要缺点是，它们忽略了光的多重散射对天空外观的影响。

为了降低射线行进的成本，包括多重散射，提出了拟合真实测量[PSS99]或使用光谱信息[HW12]生成的参考的分析模型。这些模型评估起来非常快，并从一个简单的参数化中获益：例如，一个单一的浊度值被用来表示空气中气溶胶的数量，从而产生一个看起来更密集的大气。然而，它们仅限于从地面观看和这些参数所适合的单一大气类型。例如，当模型与地球天空相匹配时，就不可能渲染火星天空。

更先进的模型已经被提出，用于渲染多次散射的大气效应，从地面到空间的视图。Nishita [NDN96]提出将参与介质细分为体素，并模拟它们之间的能量交换。已经提出了更经济的删除体素表示的模型：它们存储集成的结果，这些结果可能存储在昂贵的查找表中，可以在GPU上运行时轻松查询。这些lut可以在运行时（根据视图、太阳和世界信息）对每像素进行采样，以计算透射率和散射亮度。布鲁内顿和Neyret [BN08]提出了一个4D LUT，而Elek [Ele09]放弃了一个维度，有效地忽略了行星的大气阴影，当太阳在地平线以下时是可见的。因为，在这些模型中，从观察器到网格表面的内散射被计算为减法

^y<https://www.unrealengine.com>.

在从LUT采样的两个值中，由于分辨率和参数化精度问题，视觉伪影可能出现在水平上。Yusov [Yus13]通过更好的参数化改善了这种情况，这很适用于类地大气。然而，在大气密度更大的情况下，人工制品仍然可以看到。对于这些LUT模型，多散射是通过迭代评估内散射的LUT来实现的：从之前的散射阶LUT中采样散射亮度来评估新的LUT。当所有这些加在一起，这形成最终的散射LUT，具有多个散射顺序，直到迭代计数。然而，当游戏需要更新其大气属性时，更新这些lut是很麻烦的。g. 由于天气条件的变化或匹配艺术的方向。时间切片更新是可能的，但这将导致太阳运动和天空颜色之间的视觉延迟[Hi116]。基于lut的模型有在线的源代码，并且已经成功地地在一些游戏[Bau19]中使用。更进一步，布鲁内顿[Bru17a]广泛地讨论了所有这些模型，并比较了它们的优点和局限性。

渲染大气的挑战之一是表示由于丘陵和山脉造成的体积阴影。它可以依赖于外极线[Yus13]、阴影体积[Bru17b]，或者从阴影映射中挤出的阴影体积挤压网格的变体。这些技术速度很快，但只能表示来自不透明网格的尖锐阴影。它们将无法渲染由云参与媒体或太阳盘区域光影半影造成的软阴影。在这个领域，光线行进在捕捉这些柔软的细节方面仍然有明显的优势。

3. 参与媒体渲染

可以使用光线行进或路径跟踪来渲染参与的媒体。在这两种情况下，它都涉及到使用一个材料参数化来表示由辐射传输方程所描述的参与介质[FWKH17]。在这个框架中，对于一个给定的位置，并考虑一束光沿着一个方向移动，每波长的吸收 σ_a 和散射 σ_s 系数(m^{-1})

分别表示沿着一个方向被吸收或散射的辐射的比例。消光系数 $\sigma_t = \sigma_a + \sigma_s$ 表示吸收和外散射造成的能量损失总量。此外，当散射事件发生时，需要根据单位sr的相位函数 p 所表示的分布来确定散射方向¹。

在强实时约束下，我们的方法依赖于射线

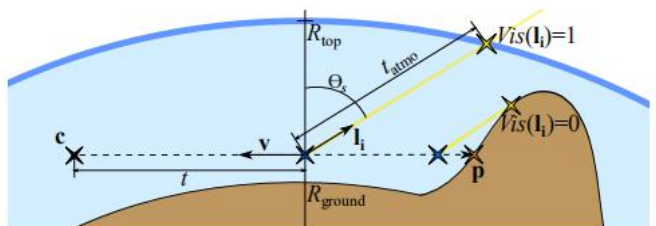


图2: 说明如何使用公式1计算参与介质中的光单次散射的示意图。

首先评估单次散射，如图2所示。它假设有一组无光定向灯，e.g. 一个太阳和月亮。它还考虑到一个虚拟行星的纯扩散响应的地面根据反照率 ρ 。它包括将散射到观测者的亮度 L 作为介质透射率 T 、阴影因子 S （ Vis 和来自大气的阴影）以及散射的函数积分 L_{scat} 沿着一个路径段使用

$$L(\mathbf{c}, \mathbf{v}) = T(\mathbf{c}, \mathbf{p}) L_o(\mathbf{p}, \mathbf{v}) + \int_{t=0}^{\|\mathbf{p}-\mathbf{c}\|} L_{scat}(\mathbf{c}, \mathbf{c}-t\mathbf{v}, \mathbf{v}) dt, \quad (1)$$

$$T(\mathbf{x}_a, \mathbf{x}_b) = e^{-\int_{\mathbf{x}=\mathbf{x}_a}^{\mathbf{x}_b} \sigma_t(\mathbf{x}) \|\mathbf{dx}\|}, \quad (2)$$

$$L_{scat}(\mathbf{c}, \mathbf{x}, \mathbf{v}) = \sigma_s(\mathbf{x}) \sum_{i=1}^{N_{light}} T(\mathbf{c}, \mathbf{x}) S(\mathbf{x}, \mathbf{l}_i) p(\mathbf{v}, \mathbf{l}_i) \mathbf{E}_i, \quad (3)$$

$$S(\mathbf{x}, \mathbf{l}_i) = Vis(\mathbf{l}_i) T(\mathbf{x}, \mathbf{x} + t_{atmo} \mathbf{l}_i), \quad (4)$$

其中 \mathbf{c} 为视图摄像机位置， \mathbf{v} 为当前位置朝向视图的方向， \mathbf{p} 为交点， t_{atmo} 射线与大气顶边界的交点距离。在 \mathbf{p} 处的亮度，例如。在这个虚拟星球的地面上进行照明。 \mathbf{L} 和 \mathbf{E}_i 是我 th 光的方向和照度（考虑定向光源）。

在本文中，我们比较了我们的新的射线行进方法与从一个路径示踪器的结果。我们的路径跟踪器是在GPU上实现的，以能够可视化在交互帧率下实时细化的结果。它在参与媒体中实现了与delta跟踪和重要性抽样的蒙特卡罗集成[FWKH17]。在估计透光率时，它还利用比率跟踪[NSJ]来实现更快的收敛。这被认为是我们的基本事实。

4. 大气模型

我们使用的大气材料模型已经在以前的论文[BN08; Bru17a]。我们专注于大地电磁行星的模拟。由一个由岩石或金属制成的固体部分组成，我们将称之为地面。行星的地面和大气层的顶部边界用等半径的球体表示。变量 h 表示地面以上的高度。对于地球，地面半径为地面 = 6360km，大气顶半径可设置为罗佩特 = 6460公里，代表一个100公里的参与媒体层。我们认为地面是一种纯扩散材料，具有均匀反照率 $\rho = 0.3$ [NAS]。当渲染大气的参与介质时，我们不考虑像[E1e09]中那样的宽光谱表示。相反，我们专注于典型的基于rgb的渲染。

大气层由几个部分组成，为了实现地球和其他行星的外观，它们都需要考虑：

- 瑞利理论代表了光与空气分子相互作用时的行为。我们假设光永远不会被吸收，只能散射在[BN08]周围。描述散射事件后光方向分布的相位函数为

$$p_r(q) = \frac{3(1+\cos(\theta)^2)}{16\pi}, \quad \text{其中 } q \text{ 为入射和出散射方向之间的夹角}.$$

表1：构成地球大气层的不同参与介质成分的系数。

类型	散射 ($10 \times 10^{-6} \text{ m}^{-1}$)	吸收 ($10 \times 10^{-6} \text{ m}^{-1}$)
瑞利	$s_r^T = 5.802; 13.558; 33.1$	$s_a^T = 0$
米	$s_m^T = 3.996$	$s_a^m = 4.40$
臭氧	$s_o^T = 0$	$s_a^o = 0.650; 1.881; 0.085$

- Mie理论代表了光在与尘埃或污染等气溶胶相互作用时的行为。

光可以被散射或被吸收。相位函数近似使用科尔内特-尚克斯相位函数[GK99] $p_m(q; g) =$

$$\frac{3}{8\pi} \frac{(1-g^2)(1+\cos(\theta)^2)}{(2+g^2)(1+g^2-2g\cos(\theta))^2}, \quad \text{其中 } g \text{ 为不对称参数}$$

确定正向和反向散射的相对强度。默认情况下， $g = 0.8$ 。请注意，它也适合使用更简单的亨耶-格林斯坦相位函数。

为简单起见，我们省略了这些相位函数的参数

在本文剩下的方程式中。我们也表示了一个各向同性的相位函数为 $p_u = \frac{1}{4\pi}$ 。

表1为各组分的散射系数和吸收系数[Bru17a]。参与的媒体跟随瑞利理论和米伊理论的高度密度分布为

$$\frac{dh}{dh} dr(h) = e \text{ 和 } dm(h) = e. \quad \text{臭氧是一种}$$

地球的特殊成分被认为是代表其大气的重要成分，因为当太阳在地平线时，它是实现天蓝色的关键[Kut13]。臭氧不会促进散射；它只吸收光。在布鲁尼顿[Bru17a; Bru17b]之后，我们将该分布表示为

$$\text{以海拔25公里为中心，宽30公里，做}(h) = \max(0; 1 - \frac{|h-25|}{15}).$$

5. 我们的渲染方法

5.1.5. 讨论：观察天空

我们现在描述天空和空中透视的视觉组件。它有助于证明我们在构建lut和使用射线行进时所做的选择。

从图3来看，类地天空的视觉频率较低，尤其是在白天：

- 瑞利散射是平滑的。
- 由于Mie散射相位函数，太阳周围的光晕对于自然界中遇到的真实相位 g 值也是相当光滑的。
- 多重散射是渲染真实图像的关键组成部分。如图3（底部一行）所示，它的视觉频率也很低。
- 在地平线上可以看到更高的频率，因为那里的大气很快变得更密集，因此光参与得更多。我们必须考虑到这一点。

大气中高频率的主要来源是由于行星的阴影（在日落时）和来自山脉的阴影遮挡了方程3中的单一散射事件。解决方案



图3: 顶部: 具有太阳、天空和空中透视图的场景, 没有(左)和(右)体积阴影。底部: 当多次散射评估(右)(左)时, 图像显示地面视图。注意: 禁用了地形上的全局照明, 以使观测结果更加可见。

我们建议可以用两种模式渲染大气: 禁用体积阴影, i. e. 利用天空视图LUT更快地渲染(见第5.3节)或启用, 即。要获得更准确但也更昂贵的渲染(请参见第7节)。

2.5 透射性LUT

当执行光线行进以整合Lscat时, 必须计算表示大气介质投射到自身上的阴影项 t 。然而, 为每个单一的散射样本执行第二次向太阳行进的射线将是昂贵的。为了加速这一点, 函数 T 被存储为一个LUT, 使用在布鲁内顿和Neyret [BN08]的第4节中描述的相同的表示。

3.5 天景LUT

考虑到构成大气的参与介质的总体频率较低(见第5.1节), 它应该足够用少量的样本射线进入它。然而, 对每个像素这样做可能是昂贵的, 特别是在高分辨率, 如4K或8K。考虑到天空的整体视觉频率较低, 我们应该能够以较低的分辨率渲染天空, 然后将其上采样到更高的分辨率。

对于一个给定的观点, 我们建议将遥远的天空渲染成一个纬度/经度纹理, 相对于行星地面上的局部向上矢量, 使地平线总是其中的一条水平线。举一个例子, 见图4, 上面代表天空, 下面代表虚拟行星地面, 中间代表地平线在其中。在第5.1节中, 我们提到了在地平线上可以看到更高频的视觉特征。为了帮助更好地表示这些结果, 我们在计算将被压缩的纹理坐标 $v \in [0; 1]$ 时, 对纬度 1 应用了一个非线性变换在地平线附近有更多的短信。使用了一个简单的二次曲线: $v = 0.5 + 0.5 * \text{符号}(1) * \frac{1+\sqrt{1-2v}}{\pi/2}$, 包含12。



图4: 白天的天空视图亮度。太阳的方向可以在左边看到, 在那里发生了米氏散射。



图5: 天空视图LUT的非线性参数化有助于将纹理细节集中在地平线上, 这在视觉上很重要。

这有效地压缩了更多的像素, 并提高了那里的细节数量。它还有助于隐藏大气以较低分辨率呈现这一事实, 如图5所示。由于低分辨率和非线性映射, 太阳磁盘没有被渲染为该纹理的一部分。它是在应用天景LUT后合成的。

4.5 空中透视图LUT

当渲染一个场景时, 空中透视图会影响到不透明的结构(e. g. 地形、山脉和建筑)和半透明的元素(例如。玻璃, 火, 或其他参与的媒体, 如云)必须渲染的一致性。因此, 与Hillaire [Hil16]类似, 我们在适合视图相机支架的体积纹理中评估了对相机的内散射和透光率(见图6)。内散射存储在RGB通道中, 而透光率存储在A通道中, 作为与波长相关的RGB透光率的平均值。

在我们的案例中使用的默认分辨率是 32×32 以及在32公里的深度范围内的32个深度切片, 这对于大多数应用程序和游戏来说已经足够了。这就是史诗游戏公司的福特尼特兹的情况, 其世界地图大小为3公里²与地球一样的程式化的大气设置。如果行星的大气层非常致密, 直到一个遥远的物体不太可见的地方, 那么

²<https://www.epicgames.com/fortnite>.

深度范围可以带到更近的视角，以便提高短范围的精度。

在评估照明后，将空中透视体积纹理应用于不透明对象，同时在屏幕上应用sky视图LUT。对于前向渲染管道中的透明元素，我们在每个顶点级别上应用空中透视图。这是因为透明元素相对于大气视觉变化通常很小。

5.5. 多次散射LUT

如第2节所述，以前的大气渲染技术[BN08; Ele09; Yus13]依赖于迭代方法来更新3D或4D LUTs，每个散射顺序进行一次迭代。这是一个可接受的解决方案，当渲染类似地球的大气时，只需要多个散射顺序为5来达到现实的天空视觉效果。然而，当渲染更厚的大气时，它很快就变得不切实际了。当较高的散射阶数对大气的外观很重要时，因此有必要在lut上迭代多次。实际上，这个复杂度 $O(n)$ （其中 n 是散射阶数）的操作在计算上太重了。当艺术家们不断更新大气属性，以匹配艺术方向或一天中不同时间的天气变化时，尤其如此。计算可以被时间分割[Hi116]，但这将导致更新延迟，这可能会影响其他系统的反应性，如全局照明或实时捕获的反射立方体映射。

我们的目标是提出一种更便宜的即时 $O(1)$ 方法，独立于散射顺序，能够评估每一帧的光多重散射贡献。保持广泛的大气设置的正确性和可信度也是一个要求，以及能够在一系列设备（从移动到高端PC）上渲染大气。最后但并非最不重要的是，我们希望我们的方法依赖于基于物理的参与媒体参数化，并实现节能。

5.1.5. 建立一个对我们的近似值的直觉

考虑到大气中整体大尺度、平均自由程长、参与介质的平稳分布，可以认为空间周围大面积内所有点到达空间点的照度 E 是相同的。从而集成亮度

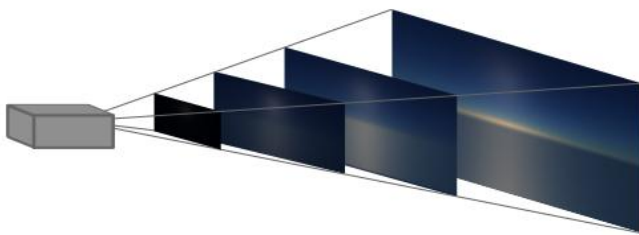


图6：相机显示空中透视LUT。这是对几个切片的内散射的可视化。

由于样本点周围的高阶光散射事件，可以通过积分周围球上的散射光，从接收相同亮度 E 的相邻散射点，同时考虑这些点之间的透射率。这种使用全局散射照度 E 作为输入，利用局部材料数据评估多次散射的想法是受到近似光在头发中的多次散射的双散射方法的启发[ZWK08]。

当光在介质中四处散射时，散射方向的分布迅速变为各向同性。为了性能，我们希望我们的多重散射LUT具有一个低维数。为了达到这个目的，我们假设根据各向同性相位函数，根据散射函数 i 。e. 没有任何首选的方向。因此，我们将忽略米伊和瑞利相位函数的设置作为多重散射近似的一部分。考虑到瑞利相位函数已经很平滑了，我们认为这是一个可接受的拟合。为了更好地直观地了解Mie散射的近似，我们请读者参考布瑟尔对散射顺序的分析[Bou08]。

此外，还发现二阶散射亮度与进一步的散射亮度之间存在相关性[HG13]。因此，我们建议评估大气中的多重散射贡献，作为到达每个样本点的二阶散射亮度的函数。

我们根据之前的结果构建了我们的方法，并将在第5.5.3节中进行深入描述。这里是对它的一个总结，以及它在评估多次散射时的近似：

- . 使用各向同性相位函数 p_u 执行阶数大于或等于2的散射事件。
- . 在我们目前遮蔽的位置附近的所有点都接收到相同数量的二阶散射光。
- . 我们计算第二散射阶贡献 L_{2nd} 次序

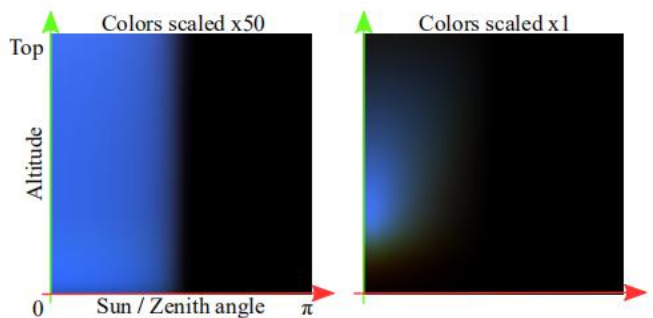


图7：在多个散射lut中存储的10 Y_{ms} 等式的可视化。左：地球设置的LUT。它是广泛均匀的，散射超过透射率。右：50次密度更大的空气，导致瑞利散射与改良的条纹-

(h) =博士。多重散射的贡献随着介质密度的增加而增加，直到透光率超过它，导致到达地面的光急剧减少。当太阳靠近地平线时，尤其如此。

和传递函数 f_{ms} （考虑到透光率和海拔高度的介质变化）从我们目前阴影的参与介质周围的位置。

- 最后，我们计算了多次散射贡献 Y_{ms} 从这些因素，模拟二阶光贡献相对各向同性的传递函数从邻域点的无限散射，回到当前的阴影位置。
- 在评估多次散射时，可见性 Vis 被忽略。
这依赖于光无论如何都会在山上散射的事实。g. 对于平均自由径较大的自然大气，能见度的影响较低。

5.2.5 LUT参数化

对于空间中的任何一个点，我们都希望能够存储和查询从一个LUT对亮度的各向同性多重散射贡献。考虑到我们认为虚拟行星是一个完美的球体，多重散射贡献是各向同性的，并且大气中介质的分布只根据海拔高度而变化，我们将这个LUT表示为一个小的二维纹理。1中的 u 参数化²是

- $u = 0.5 + 0.5 \cos(qs)$ ，其中 u 是太阳天顶角和 ω_s 表示其方向。
- $\frac{h-R_{ground}}{R_{top}-R_{ground}}v = \max(0; \min(, 1))$ ，其中样品位置 xs 在海拔高度 h 。

图7是这种lut及其参数化的一个示例。

5.5.3. 高散射阶LUT评价

考虑位置 xs 和高度 h 的样本点，我们对二阶散射亮度进行积分L2nd次序向点 xs （如图8（左）所示）使用

$$L_{2ndorder} = \int_{\Omega_{4\pi}} L'(\mathbf{x}_s, -\omega) p_u d\omega, \quad (5)$$

$$L'(\mathbf{x}, \mathbf{v}) = T(\mathbf{x}, \mathbf{p}) L_o(\mathbf{p}, \mathbf{v}) + \int_{t=0}^{\|\mathbf{p}-\mathbf{x}\|} \sigma_s(\mathbf{x}) T(\mathbf{x}, \mathbf{x}-t\mathbf{v}) S(\mathbf{x}, \omega_s) p_u E_{\mathcal{I}} dt. \quad (6)$$

在方程6中， L 术语评估具有照度的单个方向光的亮度贡献 E 和方向 ω_s ，对于与当前LUT条目匹配的位置 xs 。它还包含了地面通过 L_o 反射（根据反照率的漫反射）的亮度贡献。L2nd次序应该给二阶散射光作为亮度。但是它是使用 $E_{\mathcal{I}}$ 进行评估的：它是光照度的占位符 $E_{\mathcal{I}}$ 。虽然在这种情况下，它是一个无单位的因子 $E_{\mathcal{I}} = 1$ ，以确保L2日次序不返回亮度值，而是作为单位 sr 的传递函数¹，只有在后来与实际方向光照度乘以时才返回亮度。在方程5中，看也使用 E 但为了简单起见，我们保留了这个。

其次，我们整合了一个无单位的因子 f_{ms} 表示从所有大气中产生的能量的转移

介质周围和朝向当前阴影的样品表示 xs 为

$$f_{ms} = \int_{\Omega_{4\pi}} L_f(\mathbf{x}_s, -\omega) p_u d\omega, \quad (7)$$

$$L_f(\mathbf{x}, \mathbf{v}) = \int_{t=0}^{\|\mathbf{p}-\mathbf{x}\|} \sigma_s(\mathbf{x}) T(\mathbf{x}, \mathbf{x}-t\mathbf{v}) 1 dt. \quad (8)$$

这一点如图8（右）所示。球体上的方向积分计算为 f_{ms} ，其中细体铅字用公式8沿着每条射线进行积分。跳过对的采样是很重要的

这个方程中的阴影项 S 和相位函数，因为它在计算时已经被考虑到了L2nd次序。因此， f_{ms} 是在 $[0; 1]$ 范围内的无单位归一化转移因子。为了帮助尊重这个范围，建议使用Hillaire [Hil15]提出的方程8积分的解析解。

如上所述，我们假设到达 xs 周围的任何点的光与到达 xs 本身的散射阶大于2的光是相同的。我们可以利用这种低空间变化的假设来解析地评估多重散射的贡献。受双散射方法[ZYZK08]的启发，我们近似了无限多重散射光贡献因子 F_{ms} 作为一个几何级数，无穷和

$$F_{ms} = 1 + f_{ms} + f_{ms}^2 + f_{ms}^3 + \dots = \frac{1}{1 - f_{ms}} \quad (9)$$

最后，具有无限个散射阶数的定向光的总贡献可以计算为

$$Y_{ms} = L_{2ndorder} F_{ms}; \quad (10)$$

二阶散射贡献L2nd次序是放大器

由多重散射传递函数确定 F_{ms} 。传递函数 Y_{ms} （单位序列⁻¹），因此是简单地乘以任何方向的光照度（Lux作为 $cd \cdot sr$ 。检索对一个像素的多次散射贡献作为亮度（ $cd \cdot m^{-2}$ ）。 Y_{ms} 存储在多重散射LUT中。对于大气材料设置，此LUT适用于行星周围的任何视角和光方向。

总之，光散射方程3现在可以用我们的多重散射近似来增强，它给出了

$$L_{scat}(c; \mathbf{x}; \mathbf{v}) = ss(\mathbf{x}) \overset{\text{光}}{a} (T(\mathbf{c}, \mathbf{x}) S(\mathbf{x}, \mathbf{v}) + Y_{ms}) E_{\mathcal{I}}$$

这种简化避免了依赖于迭代方法来评估大气内的多次散射贡献。为了

我们的实时用例，集成了 F_{ms} 和L2nd次序在

单位球面采用64个均匀分布的方向来实现。有关更多的性能细节，请参阅第7节。

6. 结果

我们通过将其与两种先进的技术进行比较来验证我们的大气渲染方法：布鲁neton[Brul7a]提出的模型和体积路径追踪器。我们比较了不同的情况，并给出了每个R、G和B通道的图像均方根误差（RMSE）与地面真相的比较

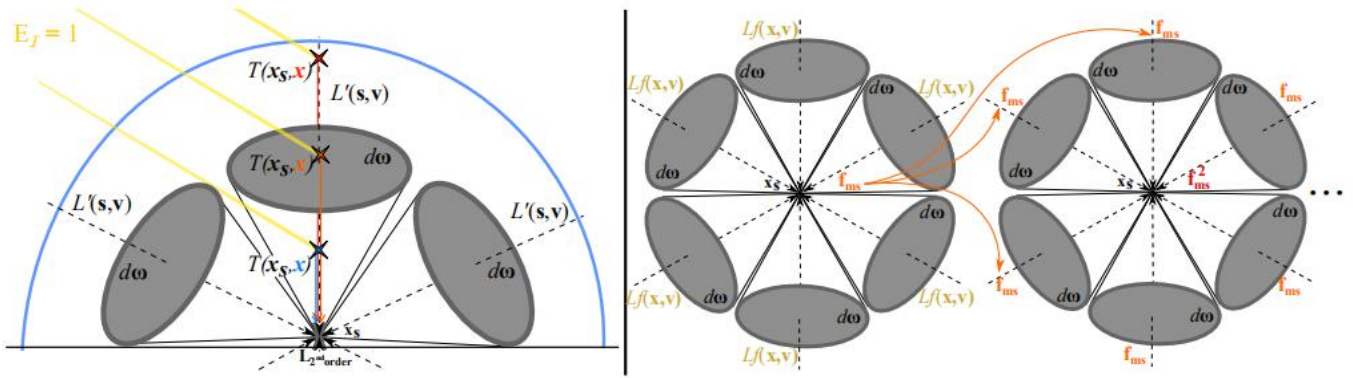


图8: 左图是 L_2 次序是由单散射 L 计算的, 在右边, 如何 f_{ms} 近似于多个散射反弹使用一个归一化传递函数 f_{ms} , 对应于方程7, 并假设一个样本点邻域接收到与样本点本身相同的能量, 对应于方程9。

路径示踪剂。我们展示了一个行星的结果使用纯黑反照率, 没有太阳盘, 以不影响RMSE测量。这个应用程序的代码是开源的吗?

首先, 我们验证了我们的模型可以忠实地呈现地球的大气层—见图9。我们提出使用单一-散射的观点, 以显示考虑多次散射时的差异。它还显示了三种模型: 布鲁内顿(B), 我们的模型(O)和参考路径示踪剂(P)。在白天, (B)和(O) RSME分别为 (1.43; 2.28; 6.07) $\cdot 10^{-3}$ 和 (0.94; 1.74; 5.07) $\cdot 10^{-3}$ —两者都非常接近参考文献(P)。对于日落的情况, 需要注意的是, (B)并不能忠实地表示由Mie散射传播的橙色。这是因为我们使用了一个单一的RGBA 4D散射LUT, 其中a代表无色的Mie散射, 而不是一个需要两个RGB 4D散射LUT的解决方案。这是在实时应用程序中使用的典型设置, 以分配更少的内存和增加

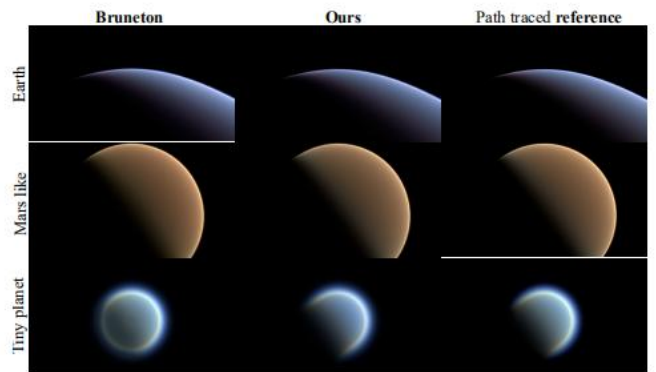


图10: 不同行星的空间视图渲染: 地球、火星和一个虚构的大气厚度和密集的小行星。

? <https://github.com/sebh/UnrealEngineSkyAtmosphere>.



图9: 用不同的技术渲染地球的大气层

在不同的条件下: 白天, 日落, 和150度的天空景观, 太阳在地平线以下, 揭示了地球在大气中投射的阴影。注意: 此图中使用各种曝光来确保视觉效果是可读的。

性能(只需要更新1个散射LUT, 它需要更少的带宽来获取LUT数据)。使用布鲁尼ton和Neyret [BN08]第4节中讨论的技巧恢复Mie散射颜色。值得注意的是, 这两种模型都能够重现地球在大气中投射的阴影中可见的苍白散射颜色——见图9的底部。

我们还比较了这些模型的精度, 以实现空间视图, 见图10。(B)模型和(O)模型都能够忠实地再现地球, 其rsme分别为 (0.58; 0.67; 1.61) $\cdot 10^{-3}$ 和 (0.95; 0.85; 1.23) $\cdot 10^{-3}$, 以及一个类似火星的行星大气层, 其大小为 (0.87; 0.97; 0.94) $\cdot 10^{-3}$ 和 (1.99; 0.91; 0.56) $\cdot 10^{-3}$. 当谈到具有浓密大气层的艺术小行星时, (B)似乎无法重现高质量的行星固体核心的体积阴影。这是由于LUT参数化, 这导致了对于天生具有高地表曲率的小行星缺乏准确性。模型(B)的这一局限性可以通过增加4D光散射LUT分辨率, 增加额外的内存和计算成本来消除。

对于地球的大气层, 据报道, 计算scat-

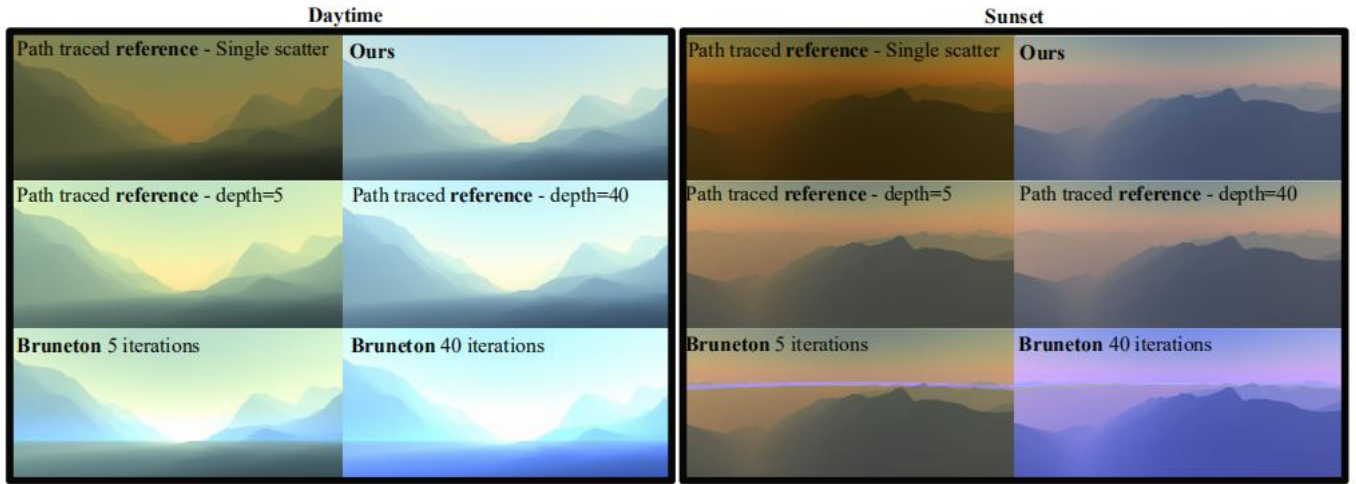


图11: 地面上的白天(左)和大气中的日落(右)视图表明,对于更密集的参与介质,考虑更高的散射顺序是很重要的。我们的方法是唯一一种可以近似于地面真实值的非迭代技术。

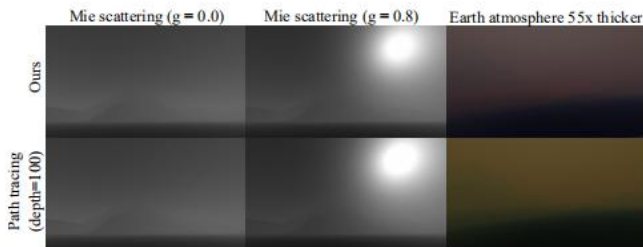


图12: 当大气变得密集时,我们的方法的局限性。左和中间: 相位 g 值越大,精度越低。右图: 致密的大气会导致不同的多重散射颜色。

只改变到第5阶就足以捕获大部分的能量[BN08],我们已经能够通过观察来证实这一点。

然而,当控制艺术家设置大气时,大气可能会变得更密集,然后考虑更高的散射阶就变得重要。虽然我们的新模型(O)会自动考虑到这一点,但模型(B)却不是这样的在这种情况下,必须有需要评估的散射顺序的迭代,这很快就变得不切实际,即使是时间切片。图11表明,对于密度较大的大气,高阶散射对于忠实地产生正确的大气颜色是至关重要的。我们的模型能够表示这种行为,而模型(B)在更高的散射阶下不能收敛到正确的颜色,甚至在数值上爆炸(图11(右))。这可能是由于在采样LUT时的精度问题,即使我们对于模型(B)散射LUT使用32位浮点表示,而不是对于模型(O)足够的16位浮点表示。

如图12所示,新的模型(O)确实有一些值得一提的问题,每个问题都是由于多重散射近似:

- 当使用非常高的散射系数时,色调可能会丢失,甚至开始漂移。
- 我们假设光的散射方向在第二次反弹后是各向同性的。这实际上是一个近似,通过我们的模型和参考路径追踪器之间的比较得到证实。0.0058仅对于Mie散射, $g=0.0$ 和 $g=0.8$, RMSE分别为0和0.039。

7. 性能和讨论

在配备了NVIDIA 1080的PC上,天空和大气透视的最终屏幕渲染是0。考虑到图9中所示的白天的情况,需要14毫秒(ms)。表2提供了由我们的方法生成的LUTs的更详细的时间和属性。最后,当分辨率为1280 720时,总渲染时间为0.31 ms。✕对于相同的视图,布鲁尼顿模型[BN08]渲染在0.22 ms,但这是没有所有的lut被更新。使用提供的[Bru17b]的代码更新所有的LUTs需要花费250 ms,其中99%的成本来自于估计多次散射所需的多次迭代。正如Hillaire [Hil16]已经显示的那样,可以在几个帧上对更新进行时间切片。然而,当评估高散射顺序时,延迟会增加,而且屏幕结果需要很长时间。

当从太空观察行星时,如图10所示,第5.3节中描述的天空视图LUT变得不那么准确,因为它的大部分被浪费在渲染空白空间上。在这种情况下,我们可以无缝地切换到屏幕上的简单光线行进。行星和大气层的渲染时间会变得更加昂贵(0.33 ms),导致总渲染成本为0.5 ms。但这通常是可以接受的,因为行星的视图集中在行星本身,所以渲染预算可能更高。

我们的技术可以从台式电脑扩展到相对较老的苹果iPhone 6s移动硬件。在这种情况下,LUT分辨率和样本计数可以缩小,而不会对产生的视觉效果产生巨大影响。本文说明了我们的设置和性能差异

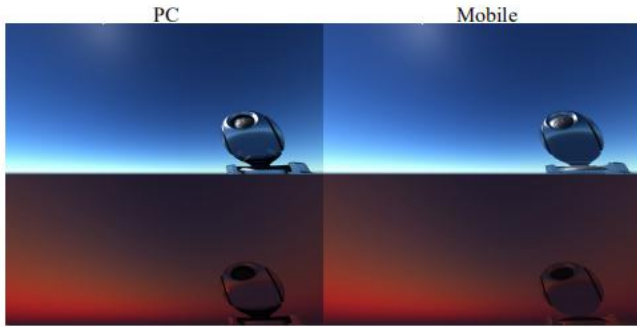


图13: PC (NVIDIA 1080) 和移动 (iPhone 6s) 渲染的视觉比较。
✖只有在白天(顶部)和日落时可以看到天空, 有5个较高的瑞利散射系数(底部)。开花、颜色分级和其他后处理效果已被禁用。

在表2中, 而视觉效果的变化如图13所示。视觉差异, 由于较低, 肉眼不明显。请注意, 我们在两个平台上都保持了相似的透光率LUT, 因为这对于确保匹配的外观很重要。如果更多的视觉差异可以换取性能, 它的质量可以在手机上进一步降低。对于Epic游戏的《堡垒之夜》, iPhone6的天空渲染成本约为1 ms。

一个重要的视觉效果是体积阴影, 例如从山脉到大气。由于大气不是一种均匀的介质, 因此不可能使用外极性采样[BCR*10]。而且也不可能使用阴影体积方法[BN08; Hoo16], 因为我们的照明方法不允许对大气中的光线路径进行这种积分采样。最后但并非最不重要的是, 这些技术不能代表云投射的软阴影: 我们必须用光线行进。与Valient [Val14]和Gjoel [GS16]类似, 我们建议使用每条射线的样本进行抖动和重新投影来结合来自之前帧的样本。抖动可以根据蓝色噪音来完成

表2: 我们的方法在个人电脑 (NVIDIA 1080) 和移动设备 (iPhone 6s) 上测量的每一步的性能。

电脑				
凹槽	分辨率	步数	渲染时间	
透射率	256 64✖	40	0.01ms	
天景	200 100✖	30	0.05ms	
空中视角	32 ³	30	0.04ms	
多散射	32 ²	20	0.07ms	
手机 (iPhone 6s)				
凹槽	分辨率	步数	渲染时间	
透射率	256 64✖	40	0.53ms	
天景	96 50✖	8	0.27ms	
空中视角	32 ² ✖16	8	0.11ms	
多散射	32 ²	20	0.12ms	



图14: 大气中的体积阴影, 从左到右图: 路径跟踪的单次散射, 路径跟踪的多重散射 (深度= 5) 和我们使用射线行进和级联阴影地图的实时方法。

和重投影可以通过时间抗锯齿 (TAA) 方法自动实现[Kar14]。这一点如图14所示。使用这种方法需要一个依赖于内容的样本计数。在这个例子中, 我们使用了32个样本, 这导致天空和大气的渲染时间上升到1.0 ms。为了降低这一成本, 也可以以较低的分辨率进行跟踪, 并暂时地重新投影和上采样结果。这已经在一些游戏引擎中使用过了, 并取得了很好的效果。使用体积阴影的结果如图1所示。

此外, 如果在第5.5节和图12节中描述的近似值是可以接受的, 我们提出的多重散射LUT也可以加速大气参与介质的路径跟踪。在这种情况下, 只需要采样单个散射事件, 例如。使用delta跟踪[FWKH17]。当这样的事件发生时, 跟踪路径可以立即停止, 在这一点上使用下一个事件估计评估单散射贡献, 剩余散射顺序的贡献可以使用多散射LUT评估。当将这种方法与我们的参考GPU路径跟踪器一起使用时, 720p帧的成本从0.74 ms下降到0.29 ms (路径深度), 如图9所示。白天的成本也从7.9 ms下降到0.6 ms, 有50个散射订单, 如图11所示。

8. 结论

总之, 我们的方法可以有效地从多个角度实时渲染天空和大气, 同时不断更新lut, 模拟光多次散射, 但不需要对每个散射顺序进行繁琐的迭代计算。这对于照明艺术家来说是很重要的, 因为他们能够实现他们的愿景, 并遵循一个项目的艺术方向, 同时模拟一天中的时间和不断变化的天气。我们已经证明, 它给出了准确的视觉结果, 即使它由于致密的大气或强烈的各向异性相函数从地面真相漂移, 结果仍然是可信的。因为它是基于物理基础的和节约能源的, 所以它不会爆炸。此外, 它还可以用于加速路径跟踪应用程序, 以渲染天空和大气。

9. 未来工作

未来的工作可能包括研究如何提高各向异性相位函数的查找表的准确性, 并支持空间变化的大气条件。我们认为, 为了提高该方法的精度[EK10]的准确性, 转向光谱渲染是很重要的。最后但并非最不重要的是, 我们

相信使用路径示踪剂和去噪器来渲染实时的天空和大气是一个很有前途的研究途径。

确认信息

我们要感谢匿名评论者的有用评论，以及Epic Games的整个渲染团队

负责审阅和校对论文，特别是纳科维茨、查尔斯·德·鲁西尔斯、格雷厄姆·维利达尔和德米特里·戴明。我们也要感谢让-塞巴斯蒂安盖伊，乔丹沃克，瑞安布鲁克斯，斯约尔德德容和维克托尔奥林曼提供的水平艺术和评估技术。最后，我们要感谢斯蒂芬·希尔对论文的校对。

参考文献

- [Bau19] BAUER, 法比安。“创造红死救赎的大气世界2：一个完整和集成的解决方案”。*实时渲染的进展, ACM签名图2019年课程*。2019 2, 9.
- [BCR*10] BARAN, ILYA, Chen, JIAWEN, RAGAN-KELLEY, jonashan, 等人。“单散射的分层体积阴影算法”。*ACM跨. 图*29.6 (2010), 178:1 – 178:10 9.
- [BN08] 布伦尼顿, 埃里克和NEYRET, 法布里斯。“预先计算的大气散射”。*欧洲图形学论文集*。2008, 1079 – 1086 2 – 5, 7 – 9.
- 安托万。“实时中对云的现实渲染”。博士论文。约瑟夫·傅里叶大学, 2008年。[URL: http://逃避.imag.fr/~Antoine.研究博士](http://逃避.imag.fr/~Antoine.研究博士)。
- 布伦顿, 埃里克。“对8个晴天模型的定性和定量评价”。*IEEE上的可视化和计算机图形学学报*23.12 (2017), 2641 – 2655 2, 3, 6.
- 布伦顿, 埃里克。预先计算好的大气散射量。2017。[URL: https://github](https://github)。预先计算的大气散射2、3、8。
- 德城堡, GILIAM和KOHEI, 西山。“Decima引擎：在照明和AA方面的进展”。*实时渲染的进展, ACM签名图2017课程*。美国纽约: ACM, 2017 2。
- ELEK, OSKAR和KMOCH, PETR。“大规模自然参与介质中的实时光谱散射”。*计算机图形学春季会议的会议记录 (SCCG)*。2010, 77 – 84 9.
- [Ele09] ELEK, 奥斯卡。“实时呈现具有多重散射的可参数化行星大气”。*CESCG* (2009) 2, 3, 5.
- [EPI18] EPICGAMES. *虚幻引擎4.19: 屏幕百分比与时间上采样*。2018年3月。[URL: https://docs.unrealengine](https://docs.unrealengine)。网络网络、引擎、渲染、屏幕百分比、索引.html 9.
- FONG, 朱利安, 宁, 马格努斯, KULLA, CHRISTO – PHER, 和HABEL, RALF。“生产卷渲染”。*ACM签名图2017课程*。2017 2, 3, 9.
- 加里·e., 托马斯和克努特, 斯坦姆尼斯。“大气和海洋中的辐射转移”。*剑桥大学. 按* (1999) 3。
- GJOEL, MIKKEL和斯文森, MIKKEL。“低复杂度, 高保真度-内部渲染”。*游戏开发者大会*。2016 9.
- 霍尔兹舒克, 尼古拉斯和加斯奎尔, 简-多米尼克岛。“半透明材料中的双散射效应和多重散射效应”。*IEEE计算机图形学与应用程序* (2013), 66–76 5。
- 希莱尔, 塞巴斯蒂恩。“基于物理和统一的体积呈现”。*实时渲染的进展, ACM签名图2015年课程*。2015 6.

希莱尔, 塞巴斯蒂安。“基于物理的天空、大气和云呈现”。*2016课程: 理论和实践中基于物理的阴影*。2016 2, 4, 5, 8.

胡布勒, 内森。“快速、灵活、基于物理的体积光散射”。*游戏开发者大会*。2016 2, 9.

HOSEK, LUKAS and WILKIE, 亚历山大。“全光谱穹顶辐射的分析模型”。*ACM跨. 图*31.4 (2012), 95:1 – 95:9 2.

[JMLH01] JENSEN, HENRIK WANN, 马施纳, 斯蒂芬R., LEVOY, MARC, 和汉拉汉, PAT。“地下光传输的实用模型”。*关于计算机图形学和交互式技术的ACM论文集*。2001, 511 – 518 5.

[Kar14] KARIS, 布莱恩。“高质量的时间超采样”。*游戏实时渲染的进展第一部分, ACM签名图2014课程*。2014, 10:1 – 10:1 9.

[Kut13] KUTZ, 彼得。臭氧的重要性。2013。[URL: http://skyrenderer.blogspot.se/2013/05/the-importanceof ozone.html](http://skyrenderer.blogspot.se/2013/05/the-importanceof ozone.html) 3.

拉加德, 塞巴斯蒂安和德罗西尔, 查尔斯。“移动冻伤到PBR”。*基于物理的理论和实践, ACM签名图2014课程*。2014 2.

NASA. 地球情况表。[URL: https://nssdc.gsfc.gov/planetary/factsheet/earthfact.html](https://nssdc.gsfc.gov/planetary/factsheet/earthfact.html) 3.

新田, 大幸, 多桥, 吉森, NAKA – MAE, 惠宏。“考虑到多重各向异性散射和天空光的云的显示”。*关于计算机图形学和交互式技术的ACM论文集*。1996, 379 – 386 2.

[NSJ] NOVAK, JAN, SELLE, ANDREW, 和JAROSZ, 哇哇哇哇。“用于估计参与媒体中的衰减的残差比跟踪”。*ACM跨. 图*33.6 (), 179:1 – 179:11 3.

新田, 大木, SIRAI, TAKAO, 蛸村, 胜美, 一弘。“考虑到大气散射的地球显示”。*关于计算机图形学和交互式技术的ACM论文集*。1993, 175 – 182 2.

O’NEIL, SEAN。“精确的大气散射”。*GPU宝石2*。2007 2.

PREETHAM, A. J., 雪莉, 彼得, 和SMITS, BRIAN。“一个实用的日光分析模型”。*关于计算机图形学和交互式技术的ACM论文集*。1999, 91 – 100 2.

[Val14] 值, MICHAL。“让杀戮地带阴影坠落形象电子科技进入下一代”。*游戏开发者大会*。2014 9.

温泽尔, 卡斯滕。“游戏中的实时大气效应”。*游戏开发者大会*。2007 2.

杨志伟, EDGARD G. *非均匀大气中的光散射*。柏林, 海德堡, 1997 5年。

[Yus13] YUSOV EGOR。“户外光散射”。*游戏开发者大会*。2013 2, 5.

ZINKE, ARNO, YUKSEL, CEM, WEBER, ANDREAS, 和KEYSER, JOHN。“在头发中快速多重散射的双散射近似”。*ACM跨. 图*27.3 (2008), 32:1 – 32:10 5, 6.