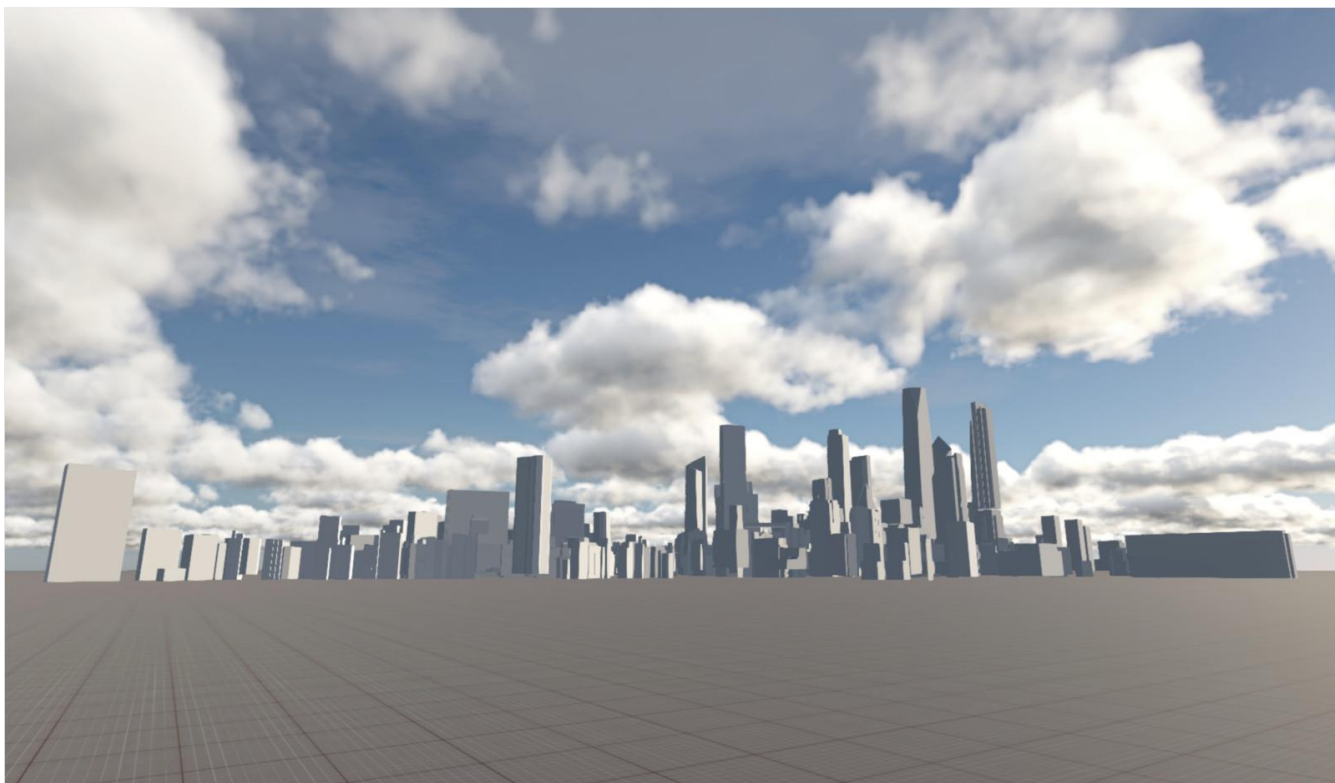




CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG



令人信服的云渲染

实时动态体积云在 Frostbite 中的实现

计算机科学硕士论文 - 计算机系统和网络

鲁里克·霍格菲尔德

哥德堡大学查尔姆斯理工大学计算机科学与工程系

瑞典哥德堡 2016

令人信服的云渲染 - Frostbite 中实时动态体积云的实现

鲁里克·赫格费尔特

© 鲁里克·霍格菲尔德 2016

监制：埃里克·辛托恩

计算机科学与工程系主考官：Ulf Assarsson

计算机科学与工程系

计算机科学与工程 查尔姆斯理工大学

SE-412 96 哥德堡

瑞典

电话 +46 (0)31 772 1000

计算机科学与工程系瑞典哥德堡，2016

抽象的

本论文描述了如何在游戏引擎Frostbite中实现实时逼真且令人信服的云。该实现的重点是渲染靠近观察者的密集云，同时仍然支持旧系统的高海拔云。新技术使用光线行进以及 Perlin 和 Worley 噪声的组合来渲染动态体积云。云被投射到圆顶中以模拟行星大气层的形状。该技术能够从不同的视角进行渲染，并且可以从大气层下方、内部和上方观看云。最终的解决方案能够在不同高度实时渲染具有许多不同云形状的真实天空。这具有完全动态的照明。

致谢

我要感谢整个冻疮团队让我感到受欢迎，并给了我与他们一起工作的机会。我要特别感谢我的两位主管冻疮, 塞巴斯蒂安·希莱尔和佩尔·埃纳尔森感谢他们在整个项目过程中的帮助和指导。

我还要感谢 马克·安德烈·洛耶在生物软件感谢他对开发和实施的帮助和支持。

最后，我还要感谢我的考官 乌尔夫·阿萨尔森和主管 埃里克·辛托恩在查尔姆斯理工大学感谢他们的帮助和支持。

内容

1	介绍	7
1.1	动机	7
1.2	目标	8
1.2.1	方法	8
1.3	报告结构	8
2	云物理	9
2.1	类型	9
2.2	行为	9
2.3	照明	11
2.3.1	吸收	11
2.3.2	散射	12
2.3.3	灭绝	12
2.3.4	透过率	13
2.3.5	排放	13
2.3.6	辐射传输方程	13
3	相关工作	14
4	执行	16
4.1	集成至Frostbite Sky模块	16
4.1.1	云模块	16
4.1.2	云着色器	17 号
4.2	噪声的产生	18
4.2.1	柏林噪声	18
4.2.2	沃利噪声	18
4.3	云纹理	20
4.4	天气纹理	20

1

介绍

实时渲染真实且令人信服的云场景一直是计算机图形学中所期望的功能。真实且令人信服的云场景不仅是参与介质中光散射的结果，也是动态云的结果，可以随着时间的推移而演变、投射阴影并与其环境相互作用。许多云渲染技术已经开发了多年，并且仍在研究中。渲染真实且令人信服的云场景仍然是一项艰巨的任务，因为云不仅具有体积和动态性，而且还需要复杂的光传输模型。

本论文中提出的云系统受到 Schneider 和 Vos 最近开发的一项技术的启发 [AN15]。该技术的主要添加和变化是不同且统一的高度信号和控制。所需的资源量也减少了。

1.1 动机

随着计算能力的增强，使用以前只适合离线渲染的新技术也变得实时可用。计算机图形学对云照明和渲染领域进行了深入研究。多年来，已经开发了许多用于渲染真实云的不同技术，但它们通常没有足够的可扩展性，无法用于游戏环境中的实时渲染。一种常见的解决方案是使用应用于天空的冒名顶替者或全景纹理。这可以产生逼真的高清云，但它是一个非常静态的解决方案。在具有动态时间和动画的框架内使用它会非常困难

邪教。这些二维解决方案也不太适合靠近观察者的云，因为它们会显得平坦并且无法真实地表示云。然而，这些技术可能适用于远离观察者的薄层云。

1.2 目标

本论文的目标是研究并实现一种实时渲染逼真且令人信服的云的技术。渲染技术应该能够产生大量形状、类型和密度不同的云，同时仍然是动态的并随着时间的推移而演变。尽管这种技术的重点是靠近观察者的浓密云层。对于艺术家来说，通过可以模拟云的形状和行为的天气系统应该很容易控制。云也应该被渲染成球形大气，这将使它们能够在地平线上弯曲。同样重要的是，云可以接收来自其环境的阴影，包括来自行星和其他云的阴影。

1.2.1 方法

该方法是研究一种不同的解决方案，可以呈现更有趣和更复杂的云场景。这是通过研究云体积渲染的最新进展，看看是否以及如何改进它们并将其应用到游戏环境中来完成的。该解决方案通过渲染时间和所需资源量进行评估。为了评估云的真实程度，使用实际照片进行比较。

1.3 报告结构

本文具有以下结构。第 2 节描述了不同的云类型、光照、它们的行为以及它们的出现方式。第 3 节向读者介绍了其他作者之前对渲染云的研究。第 4 节介绍了这些云的实现和建模。第 5 节描述了渲染。第 6 节介绍了实现实时性能所需的优化。第 7 节介绍了这种渲染技术经过优化和未经优化的结果。本节还提供了游戏内渲染和照片之间的视觉比较。第 8 节讨论并总结了此实施的结果和局限性。第 9 节涵盖了拟议的未来工作。最后，附录 A 是附加的视觉结果，附录 B 是公开控件的描述。

2

云物理

人眼看不见水蒸气，直到水在空气中凝结时才会出现云。本节简要介绍不同的云类型及其命名方式。它还涵盖了为云建模所做的行为、物理属性和简化。最后，它提供了有关参与媒体中的照明和辐射传输的部分。

2.1 类型

云可以以多种不同的形状和变化出现。大多数云类型都是根据其属性的组合来命名的。涉及海拔高度的属性是卷云对于高海拔和阿尔托适用于中海拔。属性积云用于具有蓬松外观的云。具有以下属性的云地位显示为图层。最后一个共同属性是雨云用于有降水的云。图 2.1 显示了常见的云类型及其名称。

2.2 行为

在对云进行建模时，重要的是要遵循它们的物理行为，以使它们更加真实。在不考虑物理行为的情况下渲染云不会产生令人信服和现实的结果[[国家银行+08](#)]。温度和压力是云形成和行为的关键组成部分。当水蒸气随着热量上升到较冷的大气中时，水会凝结并形成云。空气温度随着海拔的升高而降低，并且由于饱和蒸气压随着温度的升高而强烈降低，因此密度

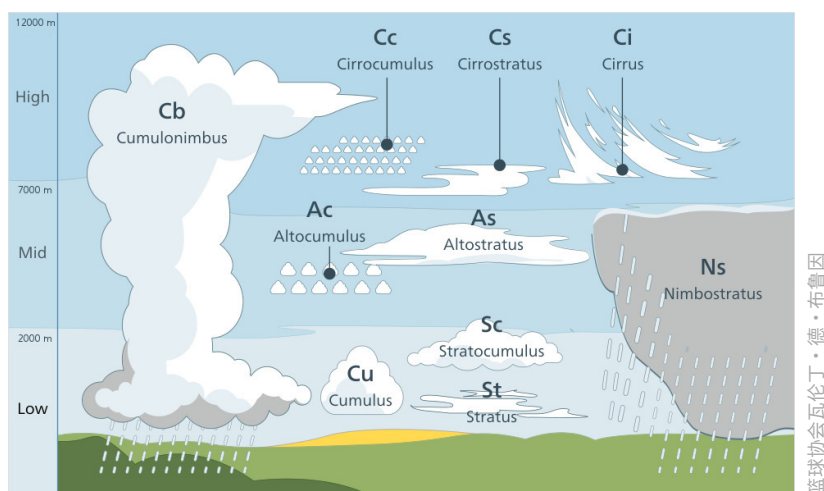


图 2.1 –常见的云类型及其名称

云通常出现在较低海拔处。雨云看起来比其他云更暗，这是雨滴尺寸较大的结果。这是因为较大的水滴吸收更多的光并散射更少的光。[MH11](#)]

风是驱动云的另一种力量，是由大气不同部分的压力差异引起的。因此，云在不同高度可能有不同的风向。由于我们的重点是靠近观察者的低空云，我们假设所有这些云都沿相同的风向移动。这使得行为变得相当复杂，因此进行了一些简化。下面的列表显示了其中一些简化。

- **云密度**: 云内部的密度随着海拔高度的增加而增加，但与云在大气中出现的位置无关。
- **风向**: 由于重点是低空云，我们只考虑一个风向。
- **液滴尺寸**: 在我们的云模型中，我们假设云始终具有相同的液滴大小，而只有密度发生变化。
- **降云**: 我们不是对具有不同液滴尺寸的降水云进行建模，而是增加这些云的吸收系数。
- **大气造型**: 我们假设大气可以被视为完美的球体而不是扁球体。

- 太阳方向: 我们假设太阳方向在大气层内可以视为平行。

2.3 照明

本节介绍穿过参与介质时的光行为。现实世界中的云没有反射光的表面。相反，光穿过它们，光子与粒子相互作用，粒子可能吸收或散射它们，从而导致辐射的变化。参与介质中的辐射率可能以四种不同的方式发生变化，这四种不同的方式如图 2.2 所示。它可能是由于吸收、内散射、外散射或发射造成的[罐子08]。

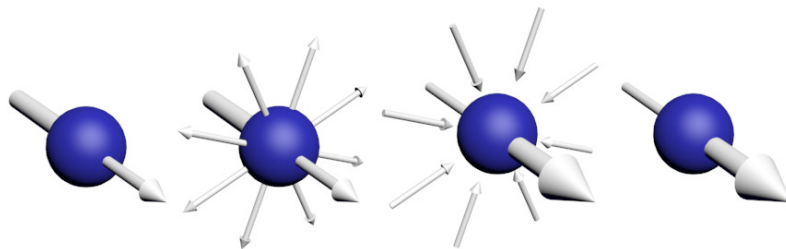


图 2.2 –光与参与媒体交互的四种方式。从左到右依次为吸收、外散射、内散射和

排放

2.3.1 吸收

吸收系数 σ_A 是光子在穿过参与介质时被吸收的概率。当光子被吸收时，它通过将光转化为热而引起辐射率的变化。由于位置吸收而减少辐射率 X 当一道光芒 L 一起旅行 ω

~是

由公式 2.1 给出。

$$e^{-\sigma_A(X) dtL(X, \omega \sim)} \quad (2.1)$$

雨云通常更暗，因为它们吸收更多的光。这是因为雨云中含有较多的较大水滴，这些水滴吸收光线的效率更高。

2.3.2 散射

辐射亮度可能因内散射而增加，也可能因外散射而减少。系数 σ_s 是光子穿过参与介质时发生散射的概率。由于内散射而增加的辐射率如公式 2.2 所示。在这个等式中 $L_{\text{我}}(x, \omega)$

\sim) 是一个相位函数，其中

确定从光方向的出散射方向 ω

\sim 。许多不同-

存在阶段功能并且适合不同类型的参与媒体。相位函数可以使光在各个方向上均匀散射为各向同性，也可以在向前和向后方向上不同地散射光。云的相位函数可能非常复杂，如图 2.3 所示，该函数用于 [国家银行+08]。

$$\sigma_s(x) L_{\text{我}}(x, \omega) = \int_{\Omega_{4\pi}} \Phi(x, \omega, \omega') L(x, \omega') d\omega' \quad (2.2)$$

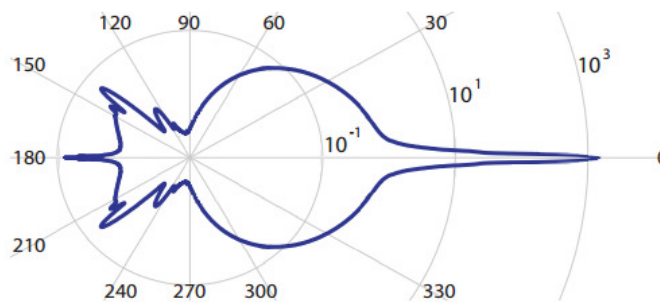


图 2.3 -用于云的 Mie 相函数图

在 [国家银行+08]

云通常是白色的，因为它们对光的散射与波长无关，这与大气散射相反，大气散射比其他云散射更多的蓝色波长。

2.3.3 灭绝

消光系数 σ_t 是穿过参与介质的光子与其相互作用的概率。光子相互作用并因此导致辐射率降低的概率是光子被吸收或向外散射的概率之和，如公式 2.3 中所述。

$$\sigma_t = \sigma_a + \sigma_s \quad (2.3)$$

2.3.4 透过率

透过率 τ 是沿直线在两点之间无障碍传播的照片数量。透射率可以使用以下公式计算 比尔-兰伯特定律如公式 2.4 中所述。

$$\tau(x_0, x_1) = e^{-\int_{x_0}^{x_1} \sigma(x) dx} \quad (2.4)$$

2.3.5 排放

发射是由于其他形式的能量转化为光而导致辐射增加的过程。由于某一点发射而增加的辐射率 L 沿着射线 ω

\sim 表示为 $L_e(x, \omega, \sim)$ 。除非有光，否则云不会发光源被放置在里面。

2.3.6 辐射传输方程

通过结合光与参与介质相互作用的四种方式的方程，可以通过能量守恒定律导出辐射传输方程。辐射传输方程如方程 2.5 所示，描述了位置处的辐射率 L 沿着有方向的射线 ω

\sim 在参与媒介内[查60]。

$$\begin{aligned} L(x, \omega) = & \int_S \tau(x, x_s) L(x_s, \omega) + \\ & \int_0^{\tau(x, x_t)} L_e(x_t, \omega) dt + \\ & \int_0^{\tau(x, x_t)} \sigma_s(x_t) L(x_t, \omega) dt \end{aligned} \quad (2.5)$$

3

相关工作

渲染云技术是一个经过深入研究的领域，最近在如何渲染逼真的云方面有许多新的突破和想法。胡纳格尔等人。提出于[\[赫赫12\]](#) 云渲染和光照研究与开发的综合调查。在这项调查中，作者比较了不同的技术并相互权衡。所涉及的一些技术包括广告牌、泼溅、体积切片和光线行进。给出了技术和适合的云类型的表格，方便读者进行比较。

最近的一项技术是 Schneider 等人开发的云系统。[\[AN15\]](#) 将在游戏中使用 *地平线零之曙光*。通过结合使用 Worley 和 Perlin 噪声以及光线行进，作者成功地在动态光照条件下实时渲染非常逼真的积云形状的云。

另一种最新的云渲染技术是由 Egor Ysov 开发的 [\[于斯14\]](#)。该技术使用预先计算的照明和粒子来渲染逼真的积云。但这种技术依赖于一种称为像素同步的功能，用于提供体积感知混合，该功能仅在英特尔高清显卡上可用[\[萨尔13\]](#)。体积感知混合可以使用光栅化有序视图来实现，这是 DirectX 12 中的一项新功能，但由于我们的实现必须在 PC、PlayStation 和 Xbox 这三个平台上运行，因此该技术不适合我们的用例。

布瑟斯等人。提议在[BNL06] 一种层状云的实时真实照明和着色技术。通过使用先进的照明模型，他们考虑了所有光路并保留各向异性行为，并使用基于物理的相位函数，他们设法以 18-40 fps 渲染逼真的云。该技术仅限于几种云类型，并且主要适用于层状云。

在 [国家银行+08]，布瑟斯等人。提出了一种模拟云中交互式多重各向异性散射的技术。通过这种技术，使用类似的方法[BNL06] 他们还设法以 2-10 fps 的速度渲染详细的积云类型云的非常逼真的照明。

4

执行

本节介绍如何实施我们的新技术并将其添加到系统中。它还涵盖了所需的资源、资源的使用方式以及生成方式。

4.1 集成至Frostbite Sky模块

Frostbite 有几个不同的天空模块可用于渲染天空。我们的实现主要集中在一个名为的天空模块上天空物理[BS14]。本节概述了新技术及其实施方式。首先我们描述了如何将云模块添加到系统中。云模块负责维护着色器的结果并向其提供输入。然后是着色器工作原理的简短概述。

4.1.1 云模块

在此实施之前渲染云的先前技术的系统图如图 4.1 所示。云是通过将称为全景和云层的静态纹理应用于天空来渲染的。这种技术可以产生令人信服的天空，但它仅限于远处的云，因为靠近观察者的云会显得平坦。

为了在靠近观察者的位置产生逼真的云，需要动态云渲染技术。静态云纹理解决方案仅适用于静态光照条件下的远处云。新的解决方案不仅要产生动态云，还要改进以前的云照明

通过处理动态光照条件进行建模。因此开发了如图4.2所示的云模块。如果场景需要，艺术家可以添加此模块。云模块渲染为纹理，然后将其提供给天空模块。该纹理每帧渲染一次，为渲染云提供动态解决方案。由于新的实现重点是渲染靠近观察者的云，旧的云层系统仍然可以用于添加高空云。

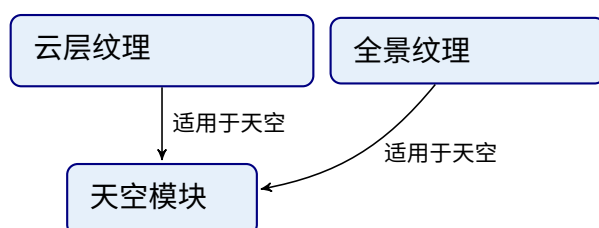


图 4.1 –显示云之前如何渲染的图表

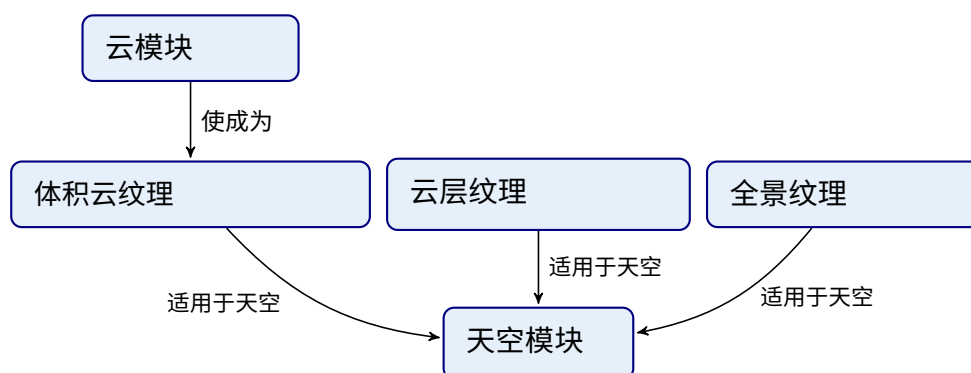


图 4.2 –显示如何添加云模块的图表

4.1.2 云着色器

我们还创建了一个与我们的云模块一起使用的着色器。该着色器是从我们的云模块调度的，并使用光线行进和不同的噪声纹理来渲染体积云。下面的第 4.2 节介绍了这些噪声，第 5 节详细介绍了着色器。

4.2 噪声的产生

本节介绍用于创建云形状的不同噪声以及它们是如何生成的。Perlin 和 Worley 噪声的组合用于创建云形状。我们在CPU上预先生成两个不同的三维纹理中的这些噪声，然后在着色器中使用它们。

4.2.1 柏林噪声

1985 年，Ken Perlin 提出了一种生成自然噪声的技术 [每85]。从那时起，这项技术就被广泛使用，为包括云在内的许多自然元素生成噪声。我们使用 Stefan Gustavson 开发的程序生成了平铺三维纹理 [古斯05]。图 4.3 是一个三维平铺纹理，该噪声映射到一个立方体。

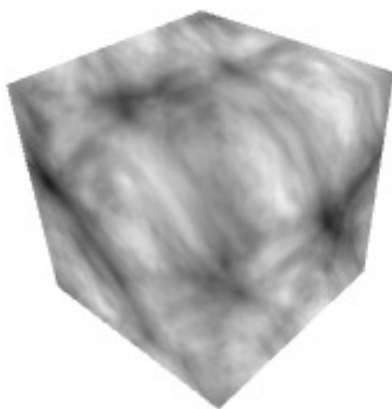


图 4.3 –应用于立方体的平铺 3D Perlin 噪波

4.2.2 沃利噪声

Stewen Worley 在 [中描述了一种产生细胞噪声的技术沃96]。图 4.4 是一个具有三维平铺沃利噪声的立方体。我们使用这种噪声类型来创建纤细和滚滚形状的云。通过反转沃利噪声，可以控制纤细和滚滚之间的外观，反之亦然。使用这种噪声创建纹理的算法可能非常简单。一种简单的方法是生成一组称为特征点的点，然后根据每个纹理元素到最近特征点的距离对其进行着色。以这种方式生成噪声会非常慢，尤其是在三维空间中。因此，如所描述的那样优化了朴素方法

在下面的步骤中。

1. 将长方体细分为大小相等的单元格。
2. 对于每个单元格，在其内部随机放置一个特征点。每个单元必须恰好有一个特征点。
3. 对于长方体内的每个点，根据到最近特征点的欧几里德距离对其进行着色。该距离是通过评估周围 26 个单元格内的特征点和当前单元格内的特征点来找到的。通过在边缘包裹单元格，纹理将在所有三个维度上平铺。

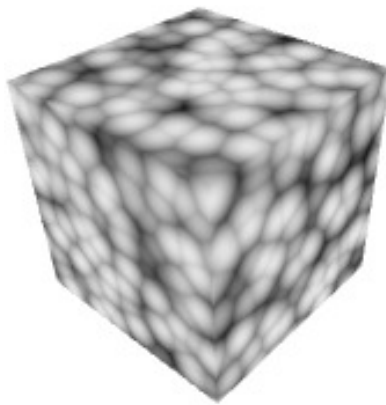


图 4.4 –应用于立方体的平铺 3D Worley 噪波

由于我们在每个单元格内放置了一个特征点，这确保了不会出现任何大的黑暗区域，如果特征点随机放置，则可能出现这种情况。在图 4.4 中，一个尺寸为 128 的立方体³使用 16 的像元大小，在立方体的任何方向上恰好产生 8 个特征点。结果是蜂窝噪声看起来是随机的，但由于特征点相距太远而没有任何大的黑暗区域。

该算法可以通过改变单元大小来生成不同倍频程的沃利噪声。倍频程是当前频率的一半或两倍，出现在间隔 2 处 n_0 。例如，生成具有 4 个八度音阶且单元大小为 2 作为起始频率的沃利噪声，使得接下来的三个八度音阶为 4、8、16。

4.3 云纹理

噪声是预先生成的并存储在两个不同的三维平铺纹理中，如表 4.1 中所述。图 4.5 是形状和细节纹理的示例。第一个三维纹理用于创建云的基本形状。它有四个通道，一个带有 Perlin 噪声，三个带有不同八度的 Worley。我们的云将在 y 轴，因此该轴的尺寸较小，以减小纹理尺寸。第二个三维纹理用于添加细节，并具有具有不同八度沃利噪声的三个通道。

表 4.1 -用于创建云形状的噪声纹理

质地	尺寸	右	G	乙	A
形状	128x32x128	柏林	沃利	沃利	沃利
细节	32x32x32	沃利	沃利	沃利	-

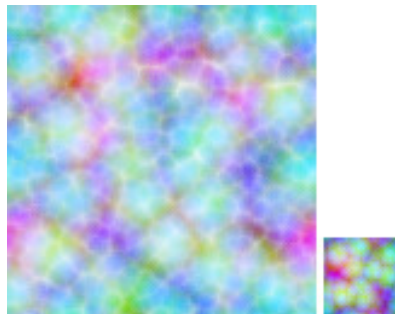


图 4.5 -带有通道的两个云纹理，如中所述
表 4.1。左：形状。右：细节

4.4 天气纹理

云由具有三个通道（称为天气纹理）的重复二维纹理控制。该纹理在整个场景中重复，并且还进行缩放以避免明显的云存在和形状图案。也用作密度的覆盖范围由该纹理中的红色通道控制。还可以通过例如根据到观看者的距离修改覆盖范围来操纵覆盖范围。这使得增加或减少地平线上的云层成为可能。绿色通道控制云的高度。值为 0 将使云具有

高度为 0，因此不可见。如果高度值为 1，则该云将具有最大高度值。蓝色通道控制云应出现的高度。值为 0 将产生等于大气起始高度的起始高度，值为 1 会使云出现在最大云层高度。这两个最大值作为我们模块中的设置公开，并且可以在运行时轻松控制。表 4.2 是天气纹理中不同通道的描述。

图 4.6 是实现过程中使用的天气纹理的示例。整体黄色调是由于覆盖范围和高度具有相似的值。除了四种不同的云之外，控制高度的蓝色设置为零。因此，使用这种天气纹理的云将出现在五个不同的高度。

表 4.2 -天气纹理的通道描述

右	G	乙	A
覆盖范围	高度	高度	-

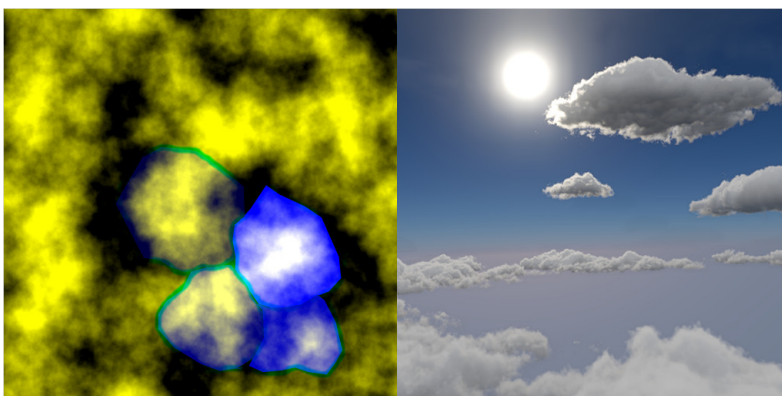


图 4.6 -天气纹理和产生的云的示例。左：天气纹理。右：使用这种天气的游戏内渲染
质地。

4.4.1 降水

较大的水滴吸收更多的光，并且在云底部的存在感更高。由于我们假设液滴尺寸均匀，因此不可能仅通过改变吸收系数来实现这一点。这甚至更加复杂，因为密度也会随着海拔高度的增加而增加，这使得底部的部分

云吸收小于顶部，这与预期相反。因此，我们的雨云会因梯度乘以光贡献而变暗。通过这种简化，所有云都可以具有相同的吸收系数，同时该技术仍然能够产生雨云和非雨云。

4.4.2 风

在现实世界中，云可以在不同的高度向不同的方向移动。我们的技术仅限于用作天气纹理偏移的单一风向。但由于我们仍然支持旧的云层系统，每一层都可以在不同方向施加风。因此，高空云可能沿与低空云不同的方向移动。

4.5 形状定义

本节介绍如何根据第 4.2 节中描述的噪声定义云形状。基础云形状由第一个三维纹理定义。Worley 噪声通道与 Perlin 噪声通道相加并相乘。

4.5.1 密度调节剂

所有云都使用相同的方法进行建模，并且密度与其在大气中出现的位置无关。首先使用 4.4 中天气纹理的覆盖值作为初始密度。然后应用第 5.4 节中介绍的高度信号。该高度信号降低了云顶部和底部的密度。然后使用两个三维纹理来侵蚀高度信号中的密度。此后应用高度梯度，降低底部的密度。高度梯度被实现为随高度的线性增加。完成任何照明计算之前的最终密度必须在范围内 $(0,1]$ 。侵蚀后为零或更低的密度可以被丢弃，因为它们不会对任何云形状产生影响。大于 1 的密度被限制为 1，以提高稳健性并使照明更加平衡。

4.5.2 详细边缘

仅从形状纹理定义云形状不足以获得详细的云。因此，较小的细节纹理用于侵蚀边缘处的云。腐蚀是通过从细节中减去噪声来执行的

带有沃利噪声的纹理。通过反转沃利噪声，可以在纤细边缘和滚滚边缘之间进行变化，或者使用两者的组合。

侵蚀量由密度决定，密度越低，侵蚀越多。侵蚀强度作为阈值实现，仅侵蚀低于该值的密度。算法 4.1 描述了密度的计算顺序。

算法 4.1 – 密度是如何计算的

```
天气      = getWeather ( 位置 .xz ) =  
密度      天气。r  
密度      *= 高度信号 (天气.gb, 位置) getCloudShape  
密度      *= (位置)  
密度 = 密度 *= 获取云尾部 (位置)  
           HeightGradient (天气.gb, 位置)
```

5

渲染

本节描述如何根据上一节中介绍的特殊密度分布来渲染和照亮云。首先，我们描述如何使用大气层以下视图中的光线行进来渲染云。然后我们描述如何轻松地将其扩展到从大气内部和空间的视图进行渲染。本节还介绍了如何实现球形气氛。然后，它描述了如何使用我们改进的高度信号在任何高度渲染任何高度的云。然后我们介绍如何应用照明以及如何添加阴影。

5.1 射线行进

光线行进是我们用于渲染云的体积渲染技术。使用此技术渲染场景时，每个像素沿着光线行进，并评估沿着该光线的每个采样点的密度、光照和阴影。更多的采样点会产生更好的结果，但反过来成本也更高。因此，有必要权衡步数和步长，以获得所需的结果。

在我们的解决方案中，我们使用的步长大小是绘制距离除以步数。绘制距离是大气深度和艺术家控制的截止距离的最小值。步骤的数量由质量选项和气氛的深度决定。这确保了即使大气被缩放到非常小或非常大，也将始终使用适当的步数。如图 5.1 所示，地平线附近的大气深度较大，但步数保持不变。这样做的效果是，靠近观察者的云具有较小的步长，从而产生更好的质量。算法 5.1 展示了光线行进是如何进行的

从行星表面观察时执行。

算法 5.1 – 雷在大气层中行进

```

入口  = 射线球体相交 (大气开始 t, 射线球体相交 (大气结 直接从眼睛) 直接从眼睛)
出口  = 束,
步长 = 距离 (入口, 步长偏移 退出) / 步骤
      = 随机 [0, 1]
t = 步长 * 步距偏移 < 脚步; +
为了 步长=0; 步 + 步骤
位置  = 入口 + 直接来自Eye * t
密度  = 获取密度 (位置)
评估 L u a t e L i g h t (密度) t +=
      s t e p L e n g t h

```

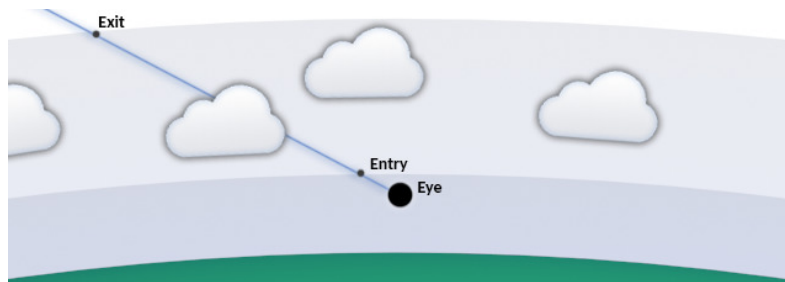


图 5.1 – 大气深度如何取决于观察方向

5.2 从内部大气和空间渲染

算法 5.1 可以轻松扩展以从大气层和太空中的视图渲染云。当我们从空间视图渲染时，我们使用图 5.1 中的出口点作为原点并进入大气层。无法使用入口点和出口点之间的距离来计算大气深度。这是因为使用空间视图时，并非所有光线都会与大气层的内部和外部边界相交。这是通过对仅与大气外部边界相交的射线使用固定大气深度来解决的。

当从大气内部渲染时，我们使用眼睛位置作为原点，而不是使用入口点。大气深度计算为从眼睛到出口点的距离以及最大截止距离。光线行进算法的自动更改由云模块处理。

5.3 圆顶投影

圆顶投影是通过使用光线-球体相交来实现的，如算法 5.2 中所述。该算法计算射线和完美球体的交点。地球的大气层形状为扁球体，赤道处的半径比两极处的半径大，但为了简化计算，我们将大气层视为完美的球体。通过使用该算法返回的位置作为入口点，云将自然地弯曲到地平线上方。

算法 5.2 - 射线球体相交

```

a = 点 (方向, b = 点 (方向方向) * 2 起始位
c = 点 (开始, 开始)      置)      * 2

判别式 = b * b - 2 * A *
          (C - 大气半径 * 大气半径) t = max (0, (-b + sqrt (判别
式)) / a) 交集 = 相机位置 + 方向
          * t

```

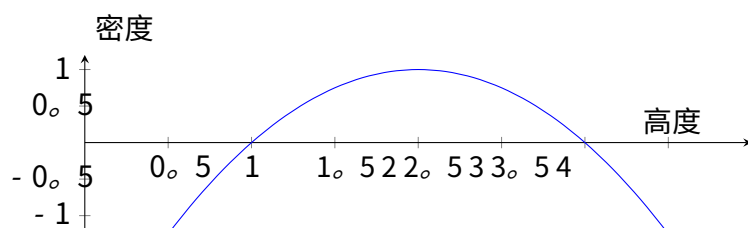
5.4 高度信号

云的高度信号以抛物线函数的形式实现，如公式 5.1 所示，其中 x 是大气中的高度， A 是云起始高度， H 是云的高度。此功能控制云出现的高度及其实际高度。高度信号函数有两个根，一个根位于云的起始高度，另一个根位于云高度。

高度信号的全局最大值被缩放通过使用这个抛物线函数云自
永远是一。然会变得更少
其顶部和底部边缘的密度。

$$\underbrace{(x-a)}_{\text{第一根}} \underbrace{(x-a-h)}_{\text{第二根}} \underbrace{-4}_{\text{规模} \cdot H} \quad (5.1)$$

图 5.2 中的图表显示了密度如何随海拔变化。在此图中起始高度 A_s 为 1 且高度 H 设置为 2，导致高度在 1 到 3 之间时密度大于 0。算法 5.3 显示了如何在着色器中实现此高度信号。

图 5.2 -高度信号 $A_s=1$ 和 $H=2$

算法 5.3 -高度信号实施

```

oneOverHeight = 1 / 高度高度 D iff
    = 高度 - 起始高度
高度信号      = 海拔高度差 * (海拔高度差 - 高度)
                * 一个超高 * 一个超高 * 4

```

5.5 照明

当射线穿过大气层时，对返回密度大于零的每个样本的照明进行评估。我们的照明模型是辐射传递函数的简化，因为我们不考虑发射。我们也不考虑云层后面的表面。需要求解的辐射传递函数的其余部分如公式 5.2 所示。

$$L(x, \omega) = \int_0^s \frac{\text{时间}_r(x, x_t) \alpha_s(x_t) L(x_t, \omega)}{s} dt \quad (5.2)$$

透射率遵循以下属性**比尔-兰伯特定律**如公式 5.3 中所述。利用这一特性，可以使用分析积分计算散射光和透射率，如算法 5.4 所示。这是使用 Hillaire 提出的散射积分方法[Sé15] 并且对于高散射值比以前的积分技术更稳定。

$$\begin{aligned} \text{时间}_r(x_0, x_1) &= e^{-\int_{x_0}^{x_1} \sigma_r(x) dx} \\ \text{时间}_r(x_0, x_2) &= \text{时间}_r(x_0, x_1) \cdot \text{时间}_r(x_1, x_2) \end{aligned} \quad (5.3)$$

算法 5.4 – 分析整合

```

样本SigmaS =      西格玛 S 散射      * 密度
样本SigmaE =      * 密度
环境=梯度          *沉淀              *全球环境颜色
S = (评估 Light ( 方向至太阳 , 位置 ) *相+
      周围的 ) *样品SigmaS
Tr = 指数 ( -样品SigmaE *一步的大小 )
      /*
          之间的分析      光/传输集成
                          脚步

      */
Sint = ( S - S *Tr ) / 样品SigmaE

散射光透过率      += 透过率 *Sint * Tr

```

5.5.1 相位功能

相位函数描述散射光的角度分布。相位函数负责云照明效果，例如一线希望、雾虹和荣耀。在我们的实现中，我们使用 Henyey-Greenstein [HG41] 相位函数如公式 5.4 所示。Henyey-Greenstein 相位函数非常适合描述云中散射光的角分布，因为它提供了非常高的前向散射峰值。这个前向峰值是在云中发现的最强的光学现象。我们考虑使用的另一个相位函数是 Cornette-Shank 提出的一个相位函数 [CS92] 并如公式 5.5 所示。该相位函数也非常适合云，但计算起来更耗时[FZZZ14]。

Henyey-Greenstein 和 Cornette-Shank 相位函数都不会像 Mie 相位函数那样产生反向散射光学现象。Henyey-Greenstein 和 Mie 相函数之间的比较如图 5.3 所示。我们没有使用米氏相位函数，因为它的振荡行为是不希望的，因为它可能会导致带状伪影。

$$p_{HG}(\theta) = \frac{1-G_2}{4\pi \cdot (1+G_2-2G \cdot \text{因斯}(\theta))^{1.5}} \quad (5.4)$$

$$p_{CS}(\theta) = \frac{3(1-G_2)(1+\text{因斯}_2(\theta))}{2(2+G_2)(1+G_2-2G \cdot \text{因斯}(\theta))^{1.5}} \quad (5.5)$$

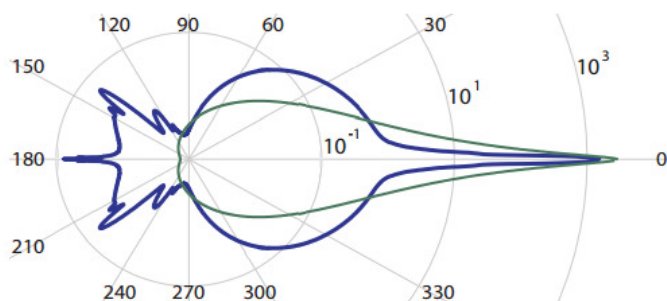


图 5.3 –云中光散射的相函数图 [国家银行+08]。蓝色：Mie，
绿色：Henyey-Greenstein 与
 $G=0$ 。 99

5.5.2 阴影

自云阴影和云间阴影是通过云内的每一步同时向太阳迈进一步来实现的，如图 5.4 所示。自云和云间阴影效果虽然对于传达深度感是必要的，但代价昂贵。向太阳行进对表现有很大影响，因为每走一步都必须多走一些步。在我们的实现中，向太阳迈出四步，步长呈指数级增加。这是因为我们希望大部分贡献来自云本身，同时仍然接收来自其他云的阴影。图 5.5 是游戏中渲染的图像，显示具有自云阴影和云间阴影的云。

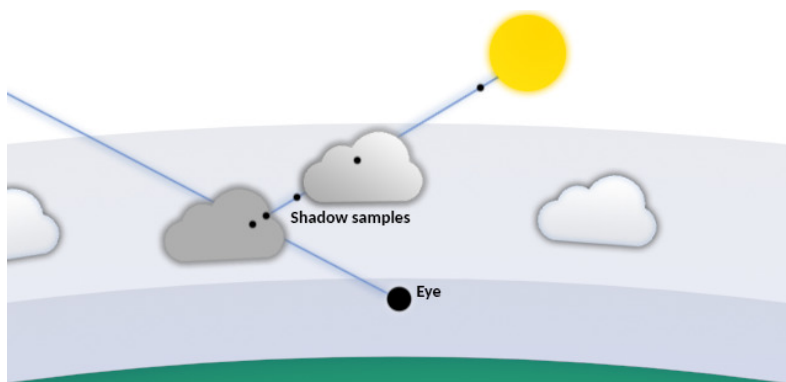


图 5.4 –呈指数增加的步长

太阳



图 5.5 – 游戏内渲染显示云间阴影

5.5.3 环境

环境光贡献是使用从云层底部到顶部的线性梯度添加的，并随着高度的增加而增加。环境贡献的强度和颜色由云模块控制。算法 5.4 描述了如何添加环境贡献。

6

优化

为了使这种云渲染技术实现实时性能，已经实施了多项优化。本节涵盖了大多数这些不同的优化。

渲染目标分辨率

将渲染目标分辨率降低到原始分辨率的一半是渲染技术实现实时性能所必需的一种优化。使用更低的分辨率可能会提供更大的性能提升，但代价是结果更模糊。

形状纹理分辨率

第一个实现在形状纹理的所有轴上具有相同的分辨率。但由于这种纹理在 X 和 z 比 y 。我们通过使用四分之一的分辨率来减少该纹理所需的资源量 y 轴。

时间上采样

通过使用时间上采样，行进时可以采取更少的步数，而不会降低云的视觉质量。本节稍后将介绍如何实现时间上采样。

提前退出

如果透射率高或低，则可以提前退出着色器（如本节后面所述）。在7.3节的渲染图像中，当透射率小于0.01时，行进终止。

6.1 时间上采样

本节介绍如何实现时间上采样以及该技术如何提高视觉质量。图 6.1 是没有时间上采样的场景，图 6.2 是使用时间上采样渲染的同一场景。两个图都使用相同的步数和长度进行渲染，唯一的区别是时间上采样。请注意启用时间上采样后带状伪影如何消失。这是因为多个帧已混合在一起，并且光线行进时的每个起始步骤都不同。下面描述了时间上采样的不同步骤。

1. 通过将步长乘以 $[0,1]$ 范围内的值来偏移射线行进算法中的起始步长，如算法 5.1 中所述。该值由云模块提供，并计算为范德科普特序列。
2. 计算世界位置 磷 使用当前逆视图投影矩阵从剪辑空间位置。

$$\text{磷} = \text{磷}_{\text{可湿性粉剂}} \cdot \text{磷}_{\text{CS}} \cdot (\text{副总裁})^{-1}$$

3. 计算， $\text{磷}'$ CS ，这是位置 磷 在上一帧中。这个做完了通过乘法 $\text{磷}_{\text{可湿性粉剂}}$ 与前一帧的视图投影矩阵。

$$\text{磷}' = \text{磷}_{\text{可湿性粉剂}} \cdot (\text{副总裁})$$

4. 在位置采样前一帧 $\text{磷}'$ 。

$$C_{\text{磷}} = \text{质地}(\text{磷})$$

5. 使用混合因子将当前帧与前一帧混合 α ，即 α 在附录 B 中。低值 α 结果会减少条带，但会花费更多时间来收敛。我们使用 5% 作为默认混合值。如果样本在我们设置的屏幕之外 α 为 1，这会导致仅渲染当前帧。为了在无法与前一帧混合时隐藏条带，我们增加了该像素的步数。

$$C_{\text{磷}} \cdot \alpha + C_{\text{磷}'} \cdot (1 - \alpha)$$

算法 6.1 –时间上采样

```

cli pSpacePos = computeClipPos ( screen enPos )
camtoWorldPos = ( cli pSpacePos , 1) * invViewProjectionMatrix
camtoWorldPos /= camtoWorldPos.w
pPrime = camtoWorldPos * 查看投影矩阵 pPrime /=
    pPrime.w
屏幕位置 = 计算屏幕位置 ( pPrime )
isOut = 任意 ( 绝对 ( 屏幕 enPos.xy - 当前 0.5 ) > 0.5 )
的 = 散射光, 是不是? 当 传输率。X
结果 = 前的 : 当前的 *  $\alpha$ 
    + 上一个 * (1- $\alpha$ )

```



图 6.1 –没有时间上采样

6.2 提前退出

执行尽可能少的计算是性能的关键。本节描述我们如何在着色器中使用早期退出点来提高性能。

6.2.1 低透过率

当穿过大气层时, 每个具有非零密度的样本都会降低透射率, 这是能量损失。因此, 随着透射率降低, 新样品对最终结果的贡献也会减少。当透射率小于0时我们停止行进。01. 这提供了巨大的性能提升, 且没有明显的视觉伪影。

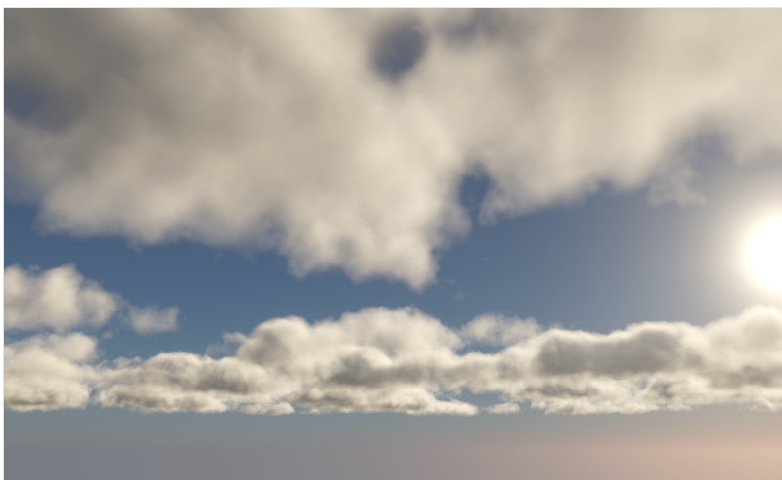
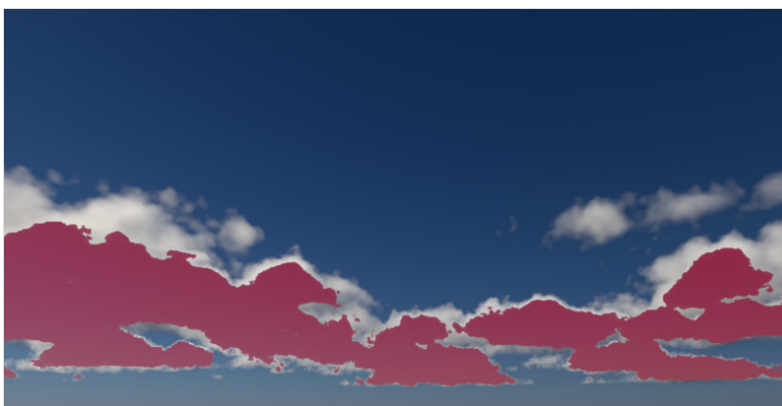


图 6.2 -通过时间上采样

图 6.3 -红色部分显示由于以下原因提前退出的时间
低透光率

6.2.2 高透过率

当透射率较高时，意味着沿该射线的样本很少具有大于零的密度。通过使用时间上采样计算的位置，我们可以从先前的结果中采样透射率，并检查该方向是否有云。如果透射率小于1.0 在这个方向上有云，但是由于风的原因，我们不能假设透射率为 1 的方向。0 没有任何云。如果所有云移动到前一帧具有高透射率的区域，则始终以高透射率退出不会渲染所有云。相反，我们会在任何情况之前提前退出

如果前一帧的透射率为 1，则行进完成。0，然后强制该像素在下一帧中行进。这可以使用方格拼贴图案自动完成，以避免一个框架比另一个框架贵很多。



图 6.4 -红色部分显示由于以下原因提前退出的时间
高透光率

7

结果

本节详细介绍了我们从该实现中获得的结果以及所需的资源、运行时性能以及最后的视觉比较。这些视觉结果集中在从行星表面渲染的云上。附录 A 中列出了大气内部和太空的其他视觉结果。

7.1 资源

表 7.1 显示了该技术所需的资源量。渲染目标所需的资源基于渲染时使用 1920x1080 的半分辨率。

表 7.1 –所需资源

质地	格式	尺寸	记忆
细节	RGB8	32x32x32	0.1MB
天气	RGB8	1024x1024	3.1MB
形状	RGBA8	128x32x128	2.1MB
渲染目标A	RGBA16	960x540	4.1MB
渲染目标B	RGBA16	960x540	4.1MB
全部的			13.5MB

7.2 性能

本节介绍使用内置性能诊断工具在 PC 上测量的运行时性能。性能评估时使用的硬件配置如表 7.2 所示。

表 7.2 –评估时使用的硬件配置
表现

硬件	发布年份
XFX Radeon HD7870	2012年
英特尔至强 E5-1650	2012年

大多数计算是在 GPU 上的像素着色器中执行的。使用内置诊断工具在 GPU 上测量执行时间。测量时使用的环境是空旷的水平面，其中只有云。图 7.1 所示为最大绘制距离恒定值为 20 000m 且观察方向向上时的执行时间。在图 7.2 中，最大绘制距离发生变化，大气深度恒定为 2 000 m，视图方向朝向地平线，以比较渲染距离如何影响运行时性能。当覆盖率较高时，几乎整个天空都被云覆盖，而当覆盖率设置较低时，只有屏幕的一小部分有云。正如预期的那样，低覆盖范围要快得多，主要是由于高透射率的提前退出。

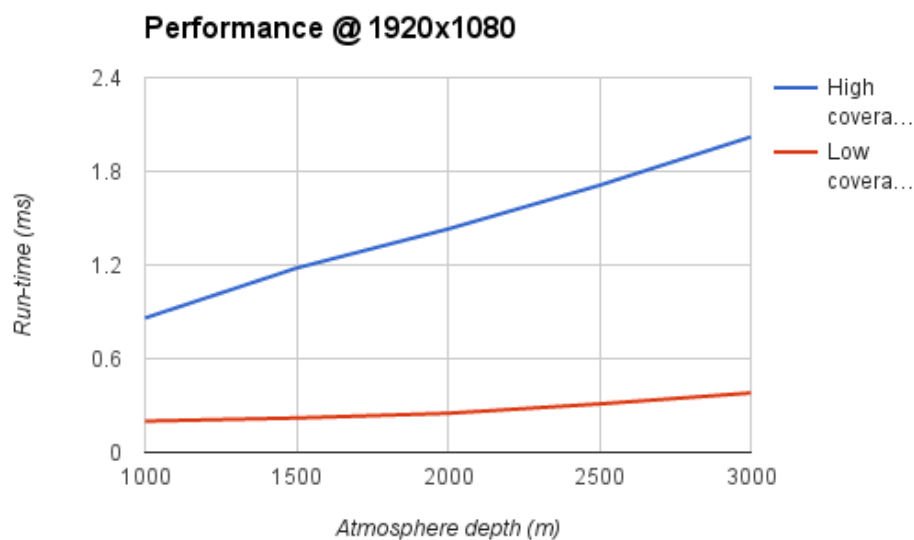


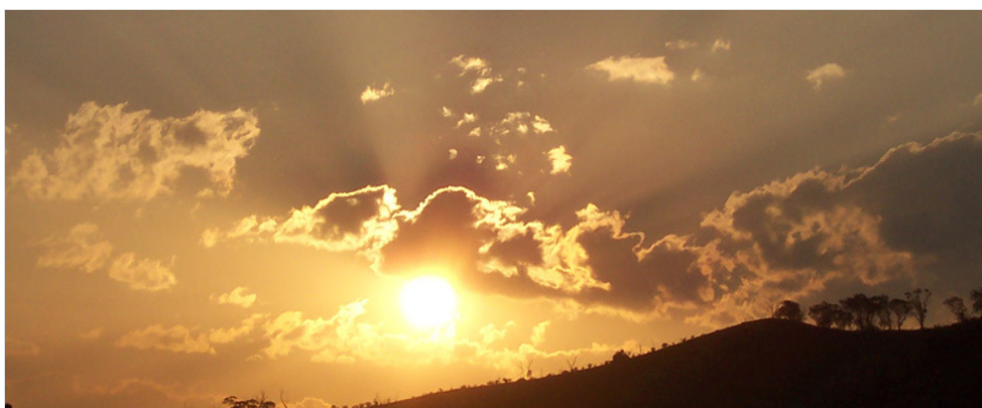
图 7.1 -运行时性能随气氛变化
深度。



图 7.2 -运行时性能作为最大渲染的函数
距离。

7.3 视觉结果

在本节中，我们将展示云渲染技术的视觉结果以及用于比较的照片。



篮球协会维基共享资源

图 7.3 -黄色日落照片



图 7.4 -游戏中的黄色日落渲染



图 7.5 –游戏中的积云渲染



篮球协会维基共享资源

图 7.6 –积云照片

8

讨论与结论

我们提出的云渲染技术可以在完全动态的光照条件下产生大量的云。该技术通过多项优化实现了实时性能，并且可以在游戏中使用，如图 A.6 所示，其中将其添加到 *战场4*。尽管该技术尚未准备好投入生产，因为它缺少上帝光线、反射视图和云等重要功能，不会在环境上投射阴影。为了实现这些功能，该技术很可能需要额外的优化。在实现过程中，我们引入了距离场和天气纹理，用于存储到最近云的距离。该距离场是使用中点圆算法预先生成的，以找到天气纹理中最近的红色像素并将其存储在不同的纹理中。距离场在某些情况下会提高运行时性能，但在其他情况下会变得更糟，因此被删除。

一种看起来非常有希望但尚未实现的优化是预计算阴影。由于我们假设与太阳的方向是平行的，因此可以预先计算阴影并将其存储在查找表中。仅当太阳移动时才需要更新该查找表。由于我们为每个样本额外采集四个朝向太阳的样本，这将大大减少样本数量。

图 4.6 中所示的天气纹理由于红色和绿色具有相似的值而具有整体黄色调。减小该纹理大小的一种优化可能是删除这些通道之一并使用相同的覆盖范围和高度。这不允许场景同时具有低和高

高密度的云，但它可能会提高运行时性能。

Henyey-Greenstein 相位函数提供了强大的前向峰值，尽管它是近似值，但对于一线希望效果来说这是必要的。积云的 Mie 相位函数是使用以下函数生成的米氏图并存储在纹理中以供着色器使用。我们的结果是背向散射光产生振荡行为并降低运行时性能。因此我们改为使用 Henyey-Greenstein，但使用 Mie 相位函数将是提供更物理准确结果的更好解决方案。

通过使用生成它来测试 Mie 相函数米氏图[拉夫45]并将其存储在查找表中，然后在着色器中使用该查找表。这给出了有希望的结果，但由于米氏相函数出现了振荡行为带。当沿着太阳方向观察时，这一点最为明显。

8.1 限制

此云渲染不支持高度重叠的云。这是因为天气纹理只存储覆盖范围、高度和起始高度。通过使用多个天气纹理可以解决此限制，但会降低运行时性能。

我们使用的相位函数会产生很强的前向峰值，但不会产生像荣耀和雾弓那样的后向散射效果。我们也只考虑太阳光和环境光，而不考虑其他光源。

8.2 道德方面

这种云渲染技术可以为许多不同的天气情况生成云，并且在游戏环境中，改变天气可能会影响玩家的情绪和行为。与计算机图形学相关的更普遍的道德方面是照片和视频操纵和伪造。

9

未来的工作

在未来的工作中，我们将优化算法并实现当前缺失的功能，例如渲染到反射视图和在环境上投射阴影。云也应该受到空中透视和雾的影响，以便更平滑地过渡到地平线。留给未来工作的另一部分是在其他平台（例如 Playstation 和 Xbox）上实现云渲染技术。

参考

- [AN15] 施奈德·安德鲁和沃斯·内森。实时体积云景地平线零黎明。在*信号图形*，2015年8月。
- [BNL06] 安托万·布托斯、法布里斯·内雷和西尔万·勒费弗尔。实时重新层状云的真实照明和阴影。在*欧洲自然现象研讨会*，2006年9月。
- [国家银行+08] 安托万·布托斯、法布里斯·内雷特、纳尔逊·马克斯、埃里克·布鲁内顿和西里尔克拉辛。云中的交互式多重各向异性散射。在*2008年交互式3D图形与游戏研讨会论文集*，I3D '08，第173–182页，美国纽约州纽约市，2008年。ACM。
- [BS14] 古斯塔夫·博代尔和爱德华·桑德伯格。 *高效、动态大气散射*。 查尔姆斯理工大学，
<http://publications.lib.chalmers.se/records/fulltext/203057/203057.pdf>，2014年。
- [Cha60] 苏布拉马尼扬·钱德拉塞卡。*辐射传输*。多佛出版社公司，1960年。
- [CS92] 威廉·M·科内特和约瑟夫·G·尚克斯。物理上合理的单散射相函数的解析表达式。*应用。选择。*，31(16)：3152–3160，1992年6月。
- [FZZZ14] 范晓雷、张利民、张兵强、查元。真实的-动态云的时间渲染。*计算机杂志*，25(3)，2014。
- [Gus05] 斯特凡·古斯塔夫森。*单纯形噪声揭秘*。林雪平大学，
<http://webstaff.itn.liu.se/~stegu/simplexnoise/simplexnoise.pdf>，2005。
- [HG41] LG Henyey 和 JL Greenstein。银河系中的漫射辐射。*天体物理学杂志*，93:70–83，1941年1月。

- [HH12] 罗兰·赫夫纳格尔和马丁·赫尔德。云照明和租金调查
德林技术。在 *WSCG* 杂志 20, 3, 第 205-216 页, 2012 年。
- [Jar08] 沃伊切赫·雅罗斯。用于光传输的高效蒙特卡罗方法
散射介质。博士论文, 加州大学圣地亚哥分校, 2008 年 9 月。
- [Lav45] 菲利普·拉文。米氏图。用于散射的计算机程序
使用米氏理论和德拜级数计算来自球体的光, <http://www.philiplaven.com/mieplot.htm>, v4.5。
- [Mh11] 郭望轩。云的颜色。香港天文台、
<http://www.weather.gov.hk/education/edu06nature/elecloudcolours.e.htm>,
2011。
- [Per85] 肯·佩林。图像合成器。在 *ACM SIGGRAPH 计算机图形-
ICS: 第 19 卷第 3 期*, 1985 年 7 月。
- [Sal13] 马可·萨尔维。像素同步。先进的渲染技术 -
电话 - 旧金山, <http://advances.realtimerendering.com/s2013/2013-07-23-SIGGRAPH-PixelSync.pdf>, 2013 年。
- [Sé15] 希莱尔·塞巴斯蒂安。基于物理的统一体积渲染
冻疮。在 *信号图形*, 2015 年 8 月。
- [Wor96] 史蒂文·沃利。细胞纹理基函数。在 *诉讼程序
第 23 届计算机图形学和交互技术年会*。第 291-294 页, 1996 年。
- [Yus14] 叶戈尔·尤索夫。使用高性能渲染逼真的积云
预先计算的照明。在 *高性能图形学报*, 2014 年。

A

额外的视觉效果

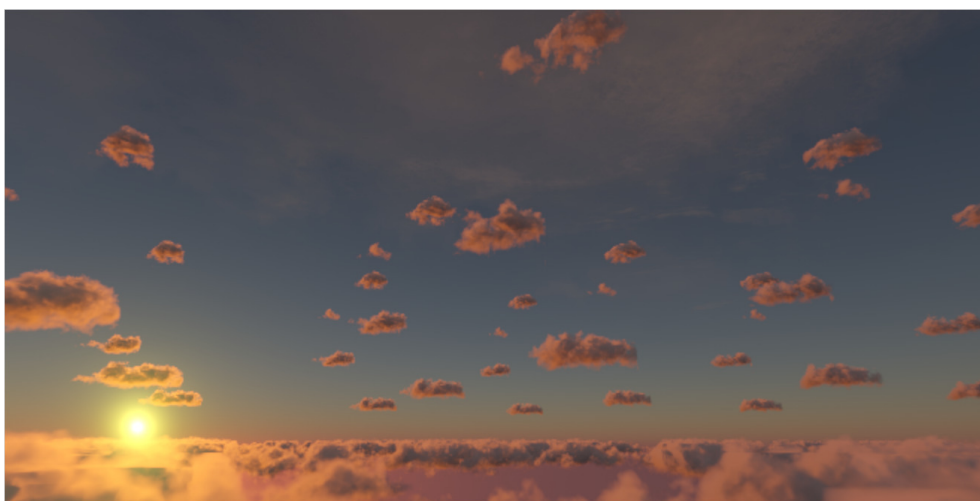


图 A.1 -起始乘数非常高的日落

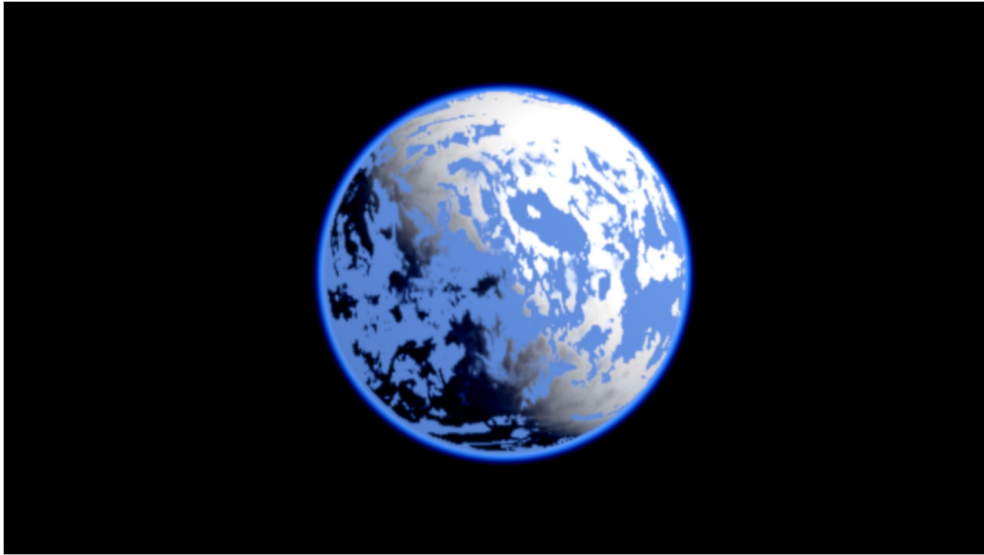


图 A.2 –从太空看云



图 A.3 –白天从内部大气层观察到的云

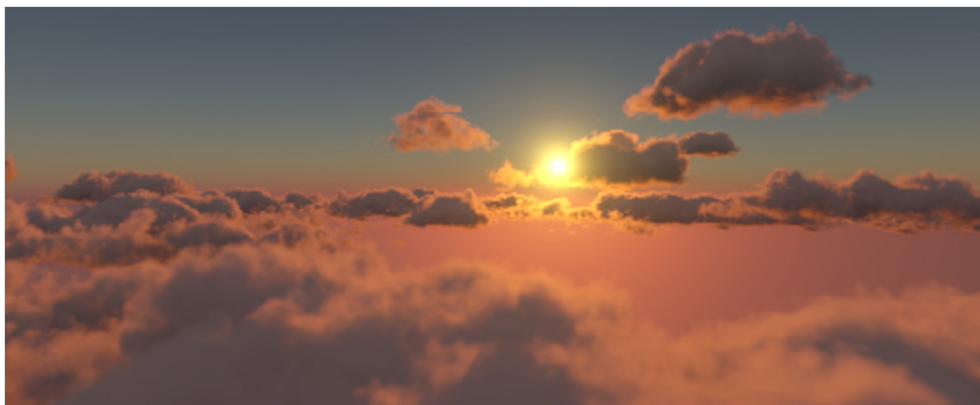


图 A.4 –从内部气氛看日落



图 A.5 –从内部气氛看日落



图 A.6 – 《战地 4》中的积云

乙

艺术家控制

本节列出了添加到我们新的云模块中用于修改云的控件及其名称和描述。这些控件按其控制的内容分组在一起。所有这些控件都可以在运行时动态更改。风和天空的控件未列出，因为它们不属于此模块。图B.1是编辑器 *弗罗斯特埃德* 显示不同的模块和云模块的公开控件。

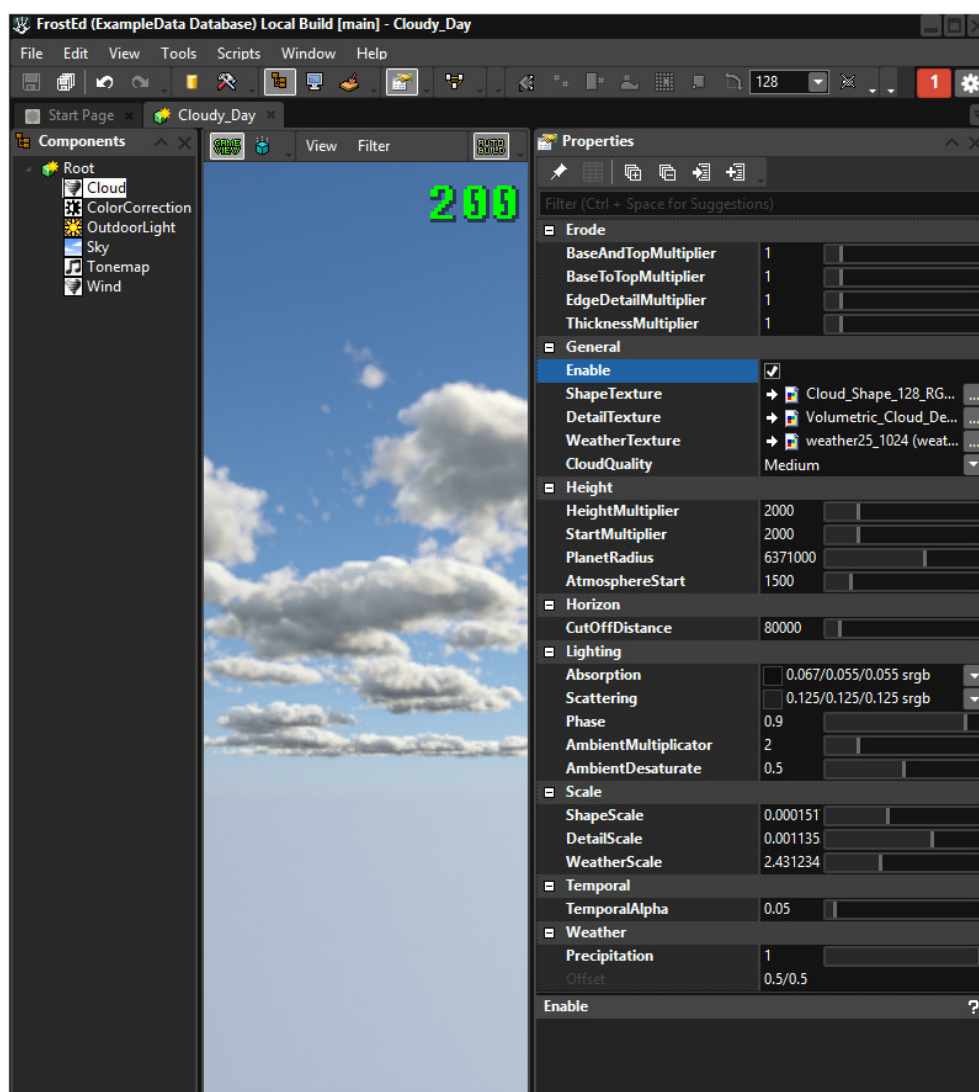


图 B.1 –编辑器显示选定的云模块并公开控制

侵蚀

- 基数和顶部乘数修改用作高度信号的抛物线函数的强度。这是该函数的极值。
- 从底部到顶部乘数修改线性高度梯度的强度。
- 边缘细节阈值用于指定应添加密度详细信息的阈值。
- 厚度倍数增加侵蚀前云形状的密度。

一般的

- 使能够启用云模块。
- 形状纹理3D 形状纹理。
- 细节纹理3D 细节纹理。
- 天气纹理2D 天气纹理。
- 云质量这是一个质量选项，可以在性能和视觉外观之间进行权衡。它调整渲染时采集的样本数量。

高度

- 高度乘数指定最大云高度（以米为单位）。
- 启动乘数指定云的最大起始高度（以米为单位）。
- 行星半径行星的半径（以米为单位）。
- 气氛启动大气层距行星表面的起始距离。

地平线

- 截止距离渲染云的距相机的最大距离。

灯光

- 吸收颜色和强度吸收系数。
- 散射颜色和散射强度系数。
- 阶段参数 G 在 Henyey Greenstein 相函数中。
- 环境倍增器环境贡献的强度。
- 环境去饱和修改环境颜色的饱和度。

规模

- 形状比例 形状纹理的缩放值。细节纹理的缩放值。
- 细节比例 缩放值。
- 天气量表 天气纹理的缩放值。

显

- 显阿尔法当前帧和前一帧之间的时间上采样的混合值。

天气

- 沉淀为云添加降水效果。
- 抵消抵消天气纹理。