

学号	姓名	论文规范性 (10)	问题分析与调研 (30)	方案创新性 (20)	实验结果分析与讨论 (40)	结课论文总成绩 (100)
21301141	杨东海	8	24	16	35	83



计算机图形学结课论文

体积云的研究与实现

Research and Implementation of Volumetric Clouds

学 院： 软件学院  
专 业： 软件工程  
学生姓名： 杨东海  
学 号： 21301141  
指导教师： 吴雨婷

北京交通大学

2024 年 6 月

## 中文摘要

体积云技术通过模拟大气中的散射和遮挡效果，能提供了更真实的视觉体验。其主要实现方法包括传统的粒子系统，以及现代化的基于体素的技术和光线步进算法。传统方法实现难度教低，但实时性能较差。而现代方法虽然计算复杂，硬件要求高，但提供了贴近真实世界的云效果。

本研究综合使用 Perlin 噪声和 Worley 噪声生成云的体积数据，结合 Beer-Lambert 定律和 Henyey-Greenstein 相函数实现光照效果。实验结果显示，现代方法在视觉效果和细节处理上优于传统方法，但需要进一步优化以提升实时性能。

**关键词：**体积云、光线步进、Perlin 噪声、Worley 噪声

## 目 录

中文摘要 .....	II
目 录 .....	III
<b>1 引言 .....</b>	<b>4</b>
<b>2 相关工作 .....</b>	<b>4</b>
2.1 传统方法 .....	4
2.2 现代方法 .....	4
2.3 比较分析 .....	5
<b>3 基本原理 .....</b>	<b>6</b>
3.1 体积渲染基础 .....	6
3.1.1 Perlin 噪声 .....	6
3.1.2 Worley 噪声 .....	6
3.2 光照模型 .....	6
3.2.1 Beer-Lambert 定律 .....	6
3.2.2 Henyey-Greenstein 相函数 .....	6
3.2.3 Ray Marching 技术 .....	7
<b>4 实验设计 .....</b>	<b>8</b>
4.1 程序架构 .....	8
4.2 体积数据生成 .....	8
4.2.1 噪声生成 .....	8
4.2.2 着色计算器 .....	8
4.3 SHADER 编写 .....	9
<b>5 实验 .....</b>	<b>10</b>
5.1 实验环境 .....	10
5.2 测试 .....	10
<b>6 结果与讨论 .....</b>	<b>11</b>
6.1 实验结果 .....	11
6.2 对比分析 .....	13
<b>7 结论 .....</b>	<b>13</b>
参考文献 .....	14

## 1 引言

体积云是通过模拟大气散射和遮挡效果来创建的虚拟物体。它们通常用于模拟天空、山脉、海洋等自然现象中的云层。体积云效果可以通过着色器和材质系统在图形渲染过程中实现。其在三维空间中具有体积，相比于简单的平面云，能提供更逼真的视觉效果。

论文主要包含相关工作介绍，体积云实现基本原理，实验方案设计，实验内容，结果与讨论的五个部分。

## 2 相关工作

### 2.1 传统方法

传统的体积云渲染方法主要依赖于粒子系统[1]，它采用大量不规则的、随机分布的具有一定生命和属性的微小粒子图元作为基本元素来描述不规则的模糊物体[1]。这种方法的优点在于能够模拟出较为自然的云层效果，但缺点是计算复杂度较高，实时渲染性能较差。

### 2.2 现代方法

现代主流的体积云渲染技术包括基于体素（Voxel）的技术<sup>[2]</sup>和光线步进（Ray Marching）算法<sup>[3][4][5]</sup>。

#### 1. 基于体素的技术：

体素化（Voxelization）技术通过将三维空间划分为固定大小的立方体（体素），并在每个体素中存储密度信息，从而实现对体积云的高效渲染。这种方法能够生成高质量的渲染图像，具有逼真的光照、阴影和材质效果，能够捕捉到复杂场景中的细节和变化。

#### 2. 光线步进（Ray Marching）：

光线步进算法是一种在 shader 中由算法生成体积云的方法。这种方法通过从相机发出射线，逐步穿过云层，根据沿途的密度信息计算出最终的颜色和亮度<sup>[4][5][6]</sup>。这种方法的优点在于能够实现高质量的视觉效果，但缺点是计算成本较高，需要优化以适应实际项目的需求。

## 2.3 比较分析

传统基于粒子系统的方法虽然简单易行，但在处理大规模场景和复杂云层结构时存在明显不足。相比之下，基于体素的技术和 **Ray Marching** 提供了更高的灵活性和真实性，尤其是在光照和阴影效果的模拟上。然而，这些现代方法在实时性能上可能不如传统方法高效，并且由于计算较为复杂，对硬件的要求也较高。

### 3 基本原理

通过 Perlin 噪声、Worley 噪声等多种噪声混合生成云的体积，利用光线步进、Beer-Lambert 定律和 Henyey-Greenstein 相函数来实现逼真的光照效果。

#### 3.1 体积渲染基础

##### 3.1.1 Perlin 噪声

Perlin 噪声（Perlin noise，又称为柏林噪声），计算通常使用插值、平滑函数，有时会在此基础上使用倍频，波形叠加（傅里叶变换）等方法对波形调整，具有较好的空间连续性<sup>[5][6]</sup>。

##### 3.1.2 Worley 噪声

Worley 噪声（Worley noise，又称细胞噪声），通过计算每个点到最近的几个特征点的距离，生成类似于细胞结构的图案。Worley 噪声特别适合模拟云的泡状结构<sup>[7]</sup>。

#### 3.2 光照模型

##### 3.2.1 Beer-Lambert 定律

描述光线在通过介质时的衰减。公式为  $I = I_0 * \exp(-\alpha * d)$ ，其中  $I$  是透射光强度， $I_0$  是入射光强度， $\alpha$  是吸收系数， $d$  是光路径长度。

##### 3.2.2 Henyey-Greenstein 相函数

用于描述光线在云层中的散射。该函数通过一个参数  $g$ （散射不对称因子）来控制散射的方向性。公式为  $P(\theta) = (1 - g^2) / (4\pi * (1 + g^2 - 2g\cos(\theta))^{3/2})$ ，其中  $\theta$  是散射角度。

### 3.2.3 Ray Marching 技术

光线步进 (Ray Marching), 又称为球体追踪 (sphere tracing), 是一种用于实时场景的快速渲染方法。模拟摄像机位置, 根据视锥体的扩张角度, 以摄像机位置为原点, 进步式发射射线, 当射线碰撞到物体之后, 返回其深度信息, 如果到视锥体的最大距离之前都没有返回, 那么可以以此判断该像素点没有对于物体, 最后根据返回的信息计算光照。

## 4 实验设计

### 4.1 程序架构

程序结构主要是 VolumetricClouds 类，在其内部封装实现函数和参数。

### 4.2 体积数据生成

基于混合的 Perlin 噪声和 Worley 噪声生成 3D 云纹理，使用计算着色器计算。具体实现如下：

#### 4.2.1 噪声生成

##### (1) Perlin 噪声

通过插值随机梯度向量上的点来生成连续的、平滑的随机噪声。

首先将定义函数将输入向量  $p$  转换为无符号整数向量  $q$ ，然后通过一系列的位操作来混合这些值，最后将结果映射回  $[-1, 1]$  的范围，然后使用此函数进行梯度插值计算，最后使用分型布朗技术（fbm）叠加多个频率和振幅来增加噪声的复杂度。

##### (2) Worley 噪声

基于点到最近特征点（通常是随机分布的点）的距离来生成的。通过叠加三个不同频率和振幅的 worley noise 增加复杂度。

##### (3) 混合

使用 remap 函数（见图）将 Perlin 噪声映射到 Worley 噪声的范围内，生成最终的 Perlin-Worley 噪声。

```
float remap(float x, float a, float b, float c, float d)
{
    return (((x - a) / (b - a)) * (d - c)) + c;
}
```

图 1 remap 函数代码

#### 4.2.2 着色计算器

局部工作组大小为  $8 \times 8 \times 8$ ，这意味着每个工作组包含 512 个线程。



纹理坐标计算：计算每个线程对应的纹理坐标，这是通过将全局调用 ID (`gl_GlobalInvocationID`) 加上 0.5 后除以纹理尺寸 (`u_Size`) 来实现的。

Perlin 噪声生成：使用 `perlin_fbm` 函数生成 Perlin 噪声，通过混合 (`mix`) 函数将其值调整为 0.5 到 1.0 之间，然后通过 `abs(perlin * 2. - 1.)` 将其转换为 “billowy” (蓬松的) Perlin 噪声。

Worley 噪声生成：通过 `worley_fbm` 函数生成三个不同频率的 Worley 噪声，分别是 `freq`、`freq * 2.0f` 和 `freq * 4.0f`。

Perlin-Worley 混合：使用 `remap` 函数将 Perlin 噪声映射到 Worley 噪声的范围内，生成最终的 Perlin-Worley 噪声。

纹理存储：将生成的噪声值（包括 Perlin-Worley 噪声和三个 Worley 噪声）存储到 3D 纹理中，每个噪声值对应纹理的一个通道。

## 4.3 Shader 编写

主要介绍体积云的片段着色器，其通过光线步进 (`ray marching`) 技术来模拟云层的形状、密度和光照效果。以下是详细介绍：

### (1) 输入、输出

`FS_OUT_Color`: 输出的颜色向量，包含了云层的最终渲染颜色。

`FS_IN_TexCoord`: 从顶点着色器传入的纹理坐标。

### (2) 主要函数

`generate_ray`: 根据屏幕上的像素位置生成一条光线。

`ray_sphere_intersection`: 计算光线与球体（代表云层边界）的交点。

`blue_noise`: 使用蓝噪声纹理来减少可见的采样模式。

`remap`: 重新映射一个值的范围到另一个范围。

`height_fraction_for_point`: 计算一个点在云层中的高度比例。

`sample_cloud_density`: 根据位置 and 高度比例采样云密度。

`sample_cloud_density_along_cone`: 沿着光锥采样云密度，用于模拟光照和阴影。

`beer_lambert_law` 和 `beer_law`: 实现 Beer-Lambert 定律，用于计算光线通过云层的衰减。

`henyey_greenstein_phase`: 实现 Henyey-Greenstein 相位函数，用于模拟光散射。

`powder_effect`: 增加粉末效果，模拟光线散射时的亮点。

`calculate_light_energy`: 计算光线能量，结合散射和衰减效果。

`ray_march`: 执行光线步进，模拟光线穿过云层的过程，计算云层的颜色和透明度。

### (3) 渲染流程

- i. 生成光线，计算光线与云层边界的交点。

- ii. 使用蓝噪声技术来随机化步进距离，减少条纹效应。
- iii. 沿着光线步进，采样云密度和光照效果。
- iv. 使用 Beer-Lambert 定律和 Henyey-Greenstein 相位函数来模拟光线通过云层的衰减和散射。
- v. 将计算得到的云层颜色与背景天空颜色混合，输出最终颜色。

## 5 实验

### 5.1 实验环境

表 1 实验环境表

服务	测试环境
CPU	CPU: i5-11300H, 内存: 16G
GPU	NIDIA GeForce MX450
软件包	OpenGL 4x , GLFW
编译器	3.28.3-msvc11
编写环境	Visual studio 2022 17.10.3
操作系统	Windows 11

### 5.2 测试

设计调试界面控制影响体积云效果的 5 个参数，具体如下：

- (1) 形状噪声比例 (Shape Noise Scale): 控制形状噪声的缩放比例。形状噪声是用来生成云的基本形状的。
- (2) 详细噪声比例 (Detail Noise Scale): 控制详细噪声的缩放比例。详细噪声用于在云的基本形状上添加更多的细节和纹理。
- (3) 详细噪声修改器 (Detail Noise Modifier): 调整详细噪声对云形状的影响程度。它可以增强或减弱详细噪声的效果，从而控制云的细节级别。
- (4) 湍流噪声比例 (Turbulence Noise Scale): 这个参数控制湍流噪声的缩放比例。湍流噪声用于模拟云中的湍流效果，这会使云的边缘和内部结构看起来更不规则和动态。
- (5) 湍流量 (Turbulence Amount): 这个参数控制湍流效果的强度。它直接影响云的不规则性和动态变化的程度。

6 结果与讨论

6.1 实验结果

基础参数如下表 2，基础参数下的渲染效果如图 2。

表 2 基础参数表	
参数名称	数值
Shape Noise Scale	0.300
Detail Noise Scale	5.500
Detail Noise Modifier	0.500
Turbulence Noise Scale	7.400
Turbulence Amount	1.000

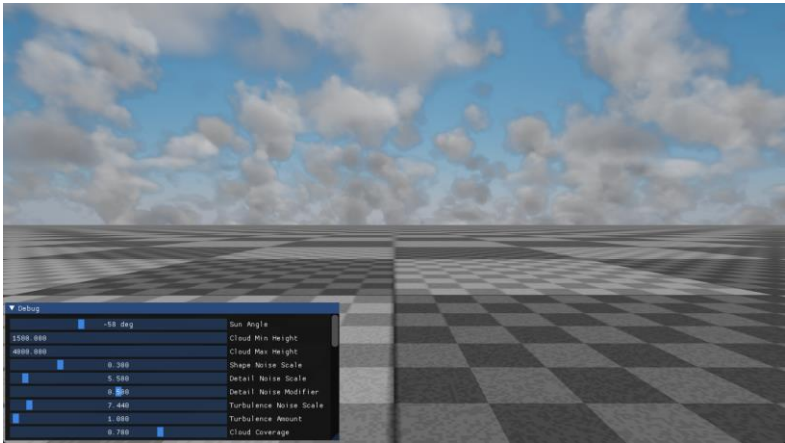


图 2 基础参数效果图

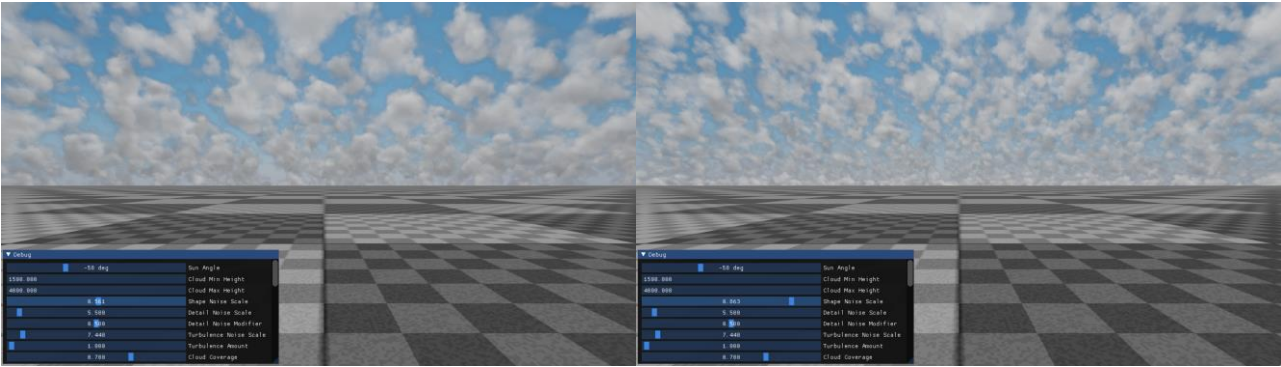


图 3 改变 Shape Noise Scale 参数效果图

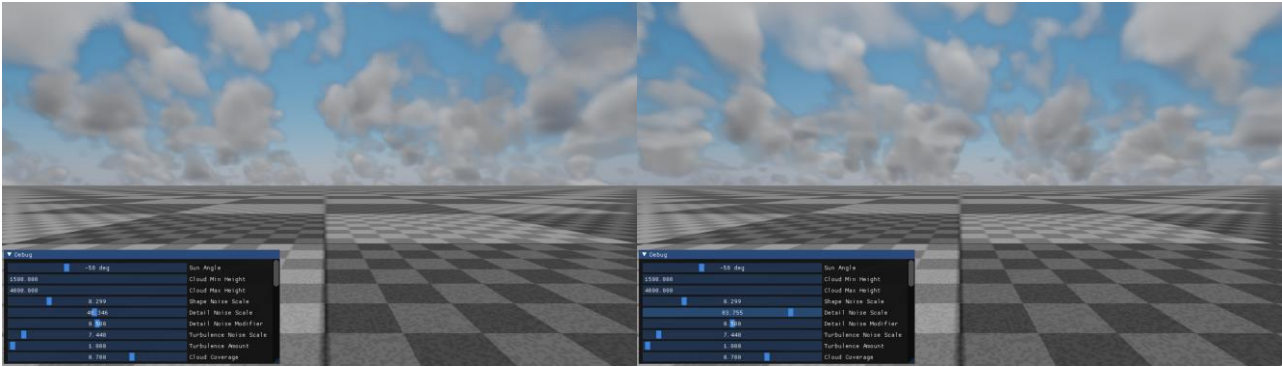


图 4 改变 Detail Noise Scale 参数效果图

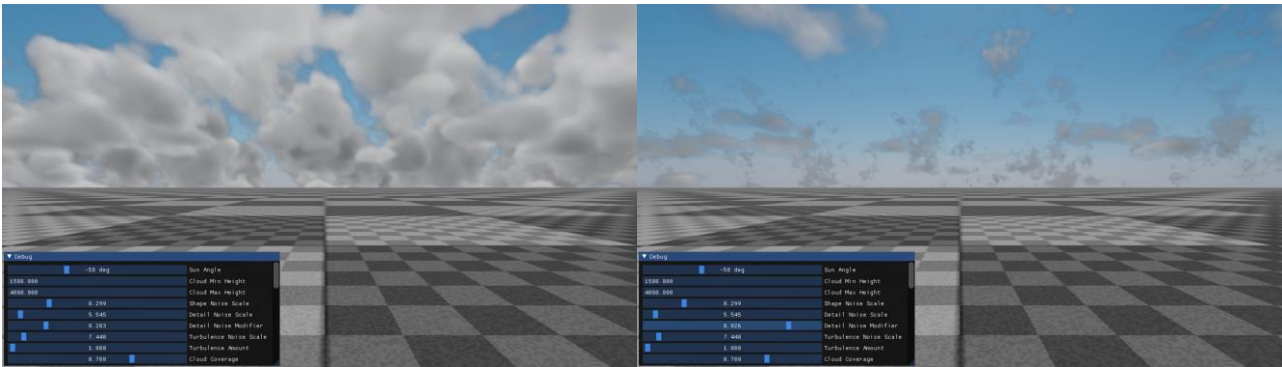


图 5 改变 Detail Noise Modifier 参数效果图

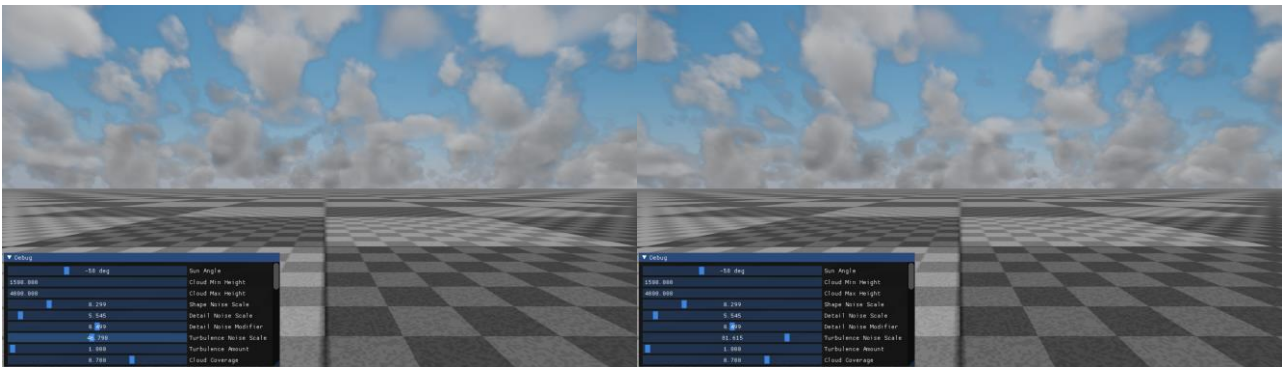


图 6 改变 Turbulence Noise Scale 参数效果图

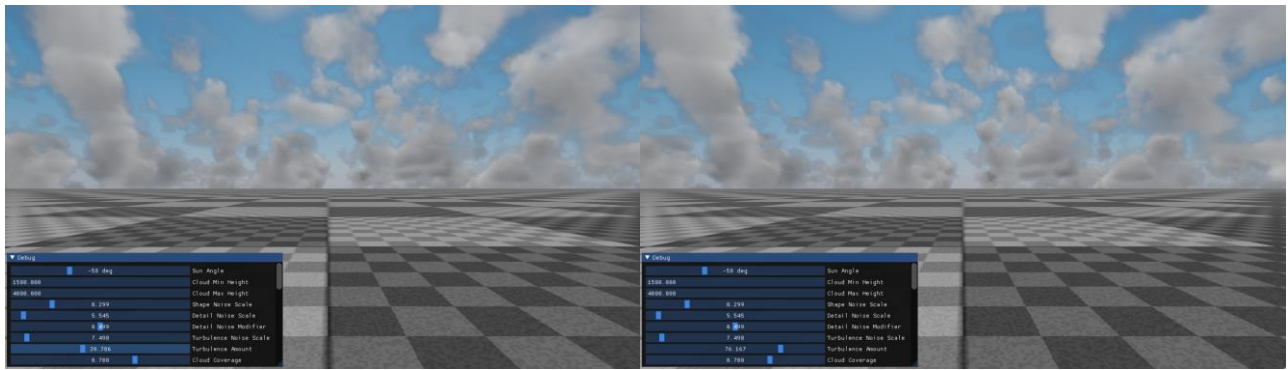


图 7 改变 Turbulence Amount 参数效果图

## 6.2 对比分析

较低或较高的 Shape Noise Scale 都会导致云块看起来更小、更细腻。这个参数直接影响云的整体形状和结构。

较低的 Detail Noise Scale 值会使云看起来更平滑，而较高的值会增加云的细节，使其看起来更复杂和真实。

较低的 Detail Noise Modifier 值会使云块更厚，较高的值会使云变薄至几乎没有。

Turbulence Noise Scale 值会修改湍流效果，从而改变云的形态，在此效果不明显。

Turbulence Amount 值会使云看起来更平静统一或更活跃多变，在此效果不明显。

## 7 结论

体积云技术能明显地提高场景渲染的真实性，通过噪声生成和光线步进等技术渲染出的体积云能提供较天空盒贴图或者模型更贴近自然环境的效果。

## 参考文献

- [1] [https://blog.csdn.net/qq\\_35097289/article/details/102395915](https://blog.csdn.net/qq_35097289/article/details/102395915)
- [2] [https://blog.csdn.net/qq\\_35097289/article/details/102395915](https://blog.csdn.net/qq_35097289/article/details/102395915)
- [3] [https://blog.csdn.net/weixin\\_46525412/article/details/126897896](https://blog.csdn.net/weixin_46525412/article/details/126897896)
- [4] <https://juejin.cn/post/7362789570217590847>
- [5] <https://blog.csdn.net/whl0071/article/details/127986949>
- [6] <https://zh.wikipedia.org/wiki/Perlin%E5%99%AA%E5%A3%B0>
- [7] [https://blog.csdn.net/qq\\_38275140/article/details/84349111](https://blog.csdn.net/qq_38275140/article/details/84349111)