# **基于Cesium的地形可视域分析实现与应用研究**

**摘要**

地形可视域分析是地理信息系统（GIS）中的一项重要空间分析功能，广泛应用于军事侦察、通信选址、城市规划等领域。本文基于Cesium平台，提出一种实时、高效的地形可视域分析方法。通过结合射线投射与地形采样技术，实现在三维地形环境中从任意观察点出发的可视区域识别与可视效果渲染。实验结果表明，该方法具备良好的实时性与可视化表现力，适用于各类可视化场景需求。

**关键词**：Cesium；地形分析；可视域分析；WebGL；三维可视化

## **1 引言**

地形可视域分析（Viewshed Analysis）指从某一地理位置出发，识别在给定地形条件下可视范围的过程。传统的可视域分析多依赖于二维栅格分析方法，但无法准确反映地形遮挡关系。随着三维GIS技术的发展，基于WebGL的可视化平台如Cesium提供了更高效、交互性更强的分析手段。

Cesium是一个开源的Web端三维地球可视化平台，支持全球地形数据、空间数据可视化与交互操作。本文利用Cesium平台，提出一种结合射线投射与地形采样的可视域分析方法，并实现其可视化表达。

## **2 相关工作**

传统GIS平台（如ArcGIS）多使用视线扫描或DEM栅格分析方式进行可视域计算，存在分辨率受限、计算效率低的问题。近年来，随着WebGL和三维可视化技术的发展，基于Cesium、Potree、Three.js等平台的三维可视域分析逐渐兴起。

研究者如Xie等人（2020）提出基于Cesium的实时射线投射方法用于三维视野计算。刘等（2022）引入地形LOD优化以提升交互效率。本文在前人工作的基础上，进一步优化了地形采样策略与渲染表现。

## **3 技术实现**

### **3.1 Cesium平台概述**

Cesium使用WebGL进行三维地球渲染，支持地形加载、3D Tiles、地理定位与场景交互。

### **3.2 需求概述**

1、实现三维求的加载，拉近显示影像图；

2、在地图上添加高亮区域面，双击地图视角可以快速的聚焦到面显示的区域；

3、添加该区域的地形图数据，显示和现实类似的高低起伏的走势；

4、在地图上显示八个黄色的点数据；

5、拖动点的位置，放置到高亮面区域位置，可以根据地形判断四周遮挡情况，可视化效果使用绿色表示可视，红色表示不可视。

### **3.2 需求技术实现**

#### **3.1.1、三维地球的实现**

**1、需要引入css和js，如下：**

 <link rel="stylesheet" href="./cesium/Widgets/widgets.css">

<script type='text/javascript' src="./cesium/Cesium.js"></script>

**2、地图实例化：**

viewer = new Cesium.Viewer("map-container", {})

**3、添加影像瓦片地图**

此处以ArcGis影像瓦片为例

viewer = new Cesium.Viewer("map-container", {

    //添加ArcGis瓦片底图图层

   imageryProvider: new Cesium.ArcGisMapServerImageryProvider({

  url:"https://services.arcgisonline.com/ArcGIS/rest/services/World\_Imagery/MapServer",

})

})

#### **3.1.2、添加高亮区域面数据**

**1、高亮区域面数据**

代码如下：

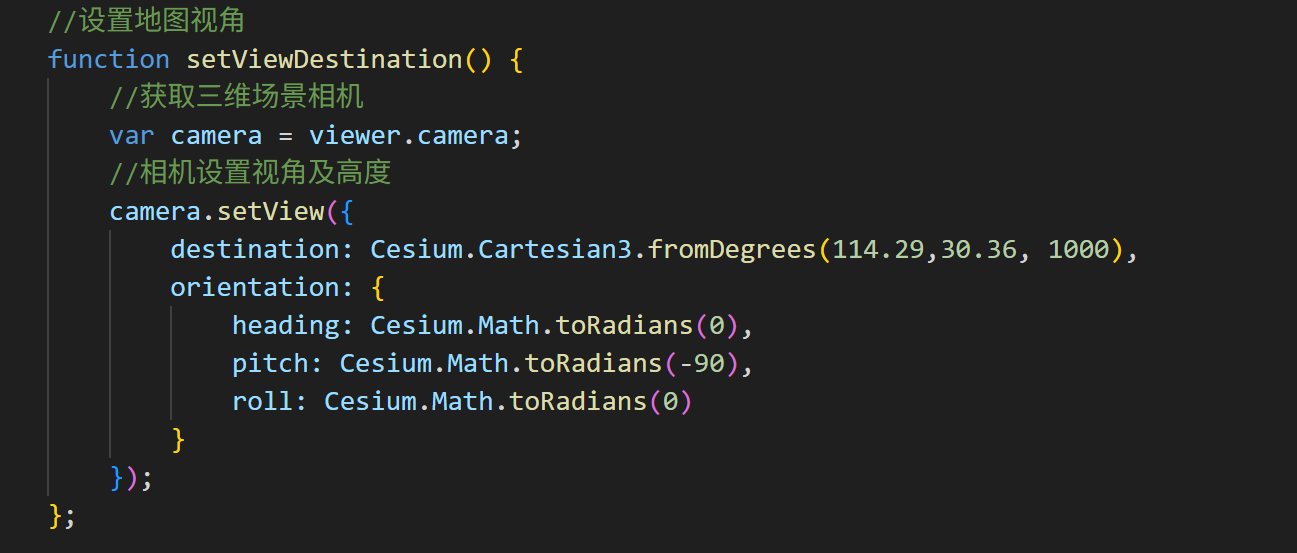


**参数介绍**

|  |  |
| --- | --- |
| **hierarchy** | 定义多边形的边界点（经纬度序列） |
| **material** | 填充多边形的颜色材质 |
| **outline** | 是否绘制多边形边界线 |
| **outlineColor** | 设置轮廓线颜色 |
| **clampToGround** | 是否将多边形贴地绘制 |

**2、地图视角聚焦**

可以通过viewer.zoomTo(polygonEntity)的方式或者以下方式：



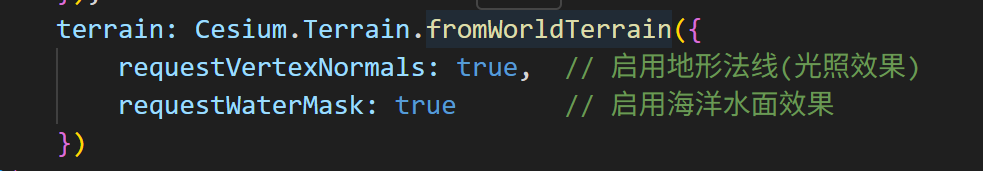
**viewer.camera**：Cesium 的相机对象，负责控制场景中观察者的位置与方向，支持平移、旋转、缩放、飞行等操作。

|  |  |
| --- | --- |
| **destination** | 相机的位置，即从哪儿看 |
| **heading** | 相机朝向水平方向的朝向 |
| **pitch** | 相机朝向绕 X 轴旋转，即俯仰角 |
| **roll** | 绕 Y 轴旋转，表示图像是否倾斜 |

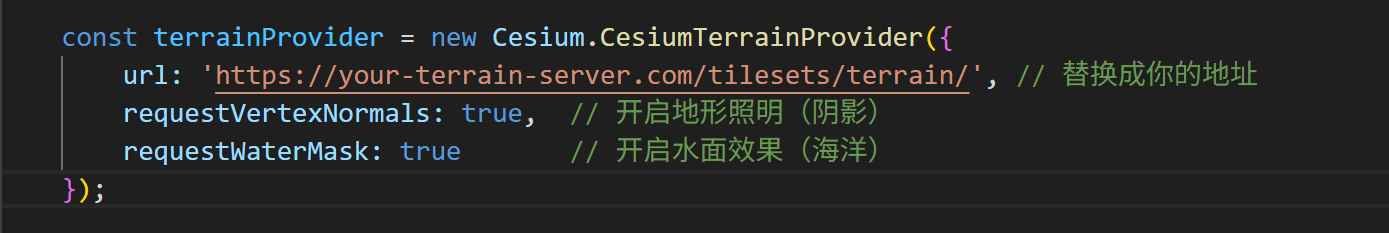
#### **3.1.3、地图上添加地形图数据**

在Cesium添加地形图方式有多多种，可以使用内置的地形，亦可以自己发布地形图片数据加载。

**1、使用内置的地形，代码如下：**

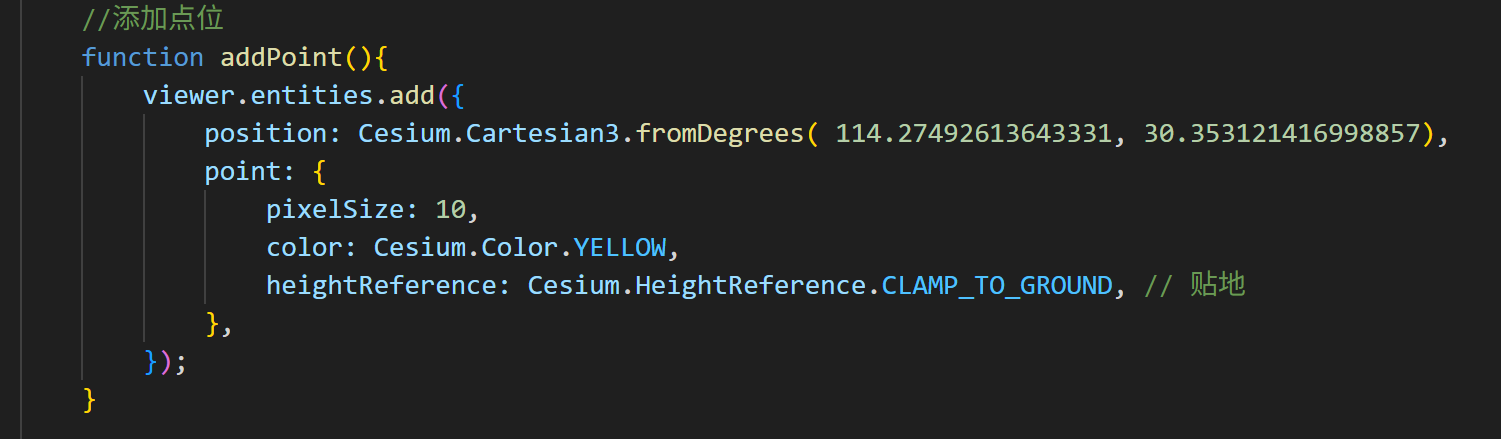


**2、自己发布地形图加载，代码如下：**



#### **3.1.4、地图上添加八个黄色的点**

实现方法如下：



**参数介绍：**

|  |  |
| --- | --- |
| **position** | **地图定位的经纬度坐标点** |
| **pixelSize** | **像素点大小** |
| **color** | **像素点颜色黄色** |
| **heightReference** | **高度参考** |

#### **3.15、点的可视域分析**

可视域分析在许多领域有着广泛的应用：

**1、军事领域**：评估观察哨、雷达站的覆盖范围，**识别视野死角和最佳观测位置**。

**2、城市规划**：分析新建筑对周围景观视野的影响，**确保重要景点的可见性**。

**3、通信网络规划：优化信号塔的位置，确保最大的信号覆盖范围。**

**4、安防监控**：评估监控摄像头的覆盖范围，**消除监控盲区**。

**5、旅游景点设计：确定最佳观景平台位置，提升游客体验。**

**6、环境影响评估：分析新项目对周围环境视觉影响的范围。**

**技术原理解析**

Cesium中实现可视域分析的核心原理主要是利用阴影映射（Shadow Mapping）技术。该技术原本用于渲染阴影，但我们可以巧妙地应用于可视域分析中：

1、创建虚拟光源相机：将观测点作为光源，设置相机的视锥体参数（如视距、水平视角、垂直视角等）。

2、生成深度图：从虚拟光源相机角度渲染场景，生成深度图（Shadow Map）。

3、可见性判断：对于每个像素点，比较其到观测点的实际距离与深度图中记录的距离，判断该点是否可见。

4、着色处理：根据可见性结果，对可见区域和不可见区域分别使用不同颜色进行着色。

**相关代码解析：**

**1、设置观测点；**

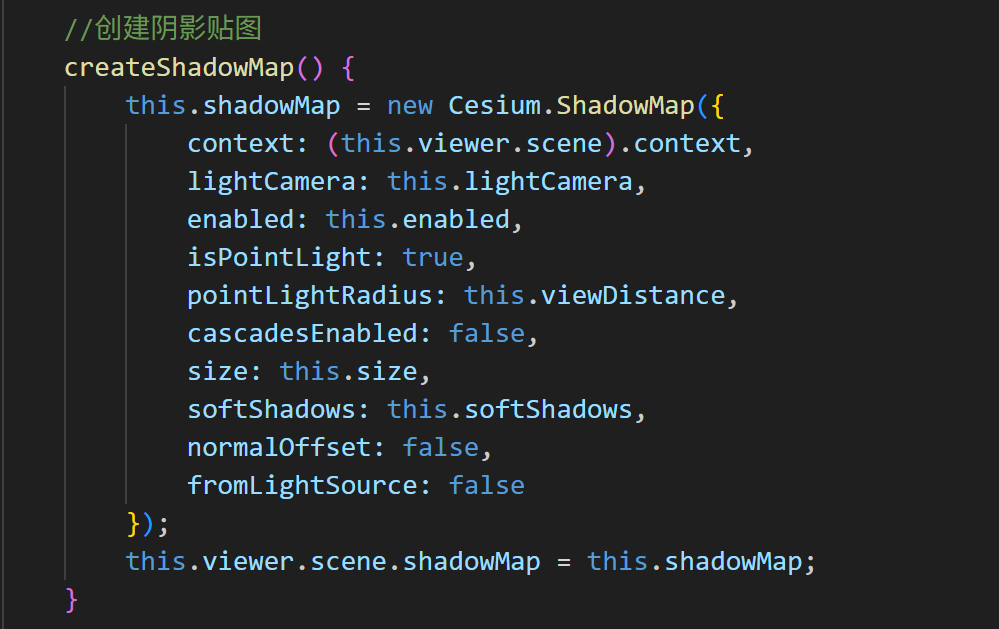
可以手动交互式的，也可以自己设定

**2、创建虚拟点光源，代码如下：**



这个方法创建一个虚拟相机作为光源，位于观测点位置，朝向观测方向，并设置视锥体参数（近平面、远平面、视场角、纵横比等）。这个相机将用于生成深度图。

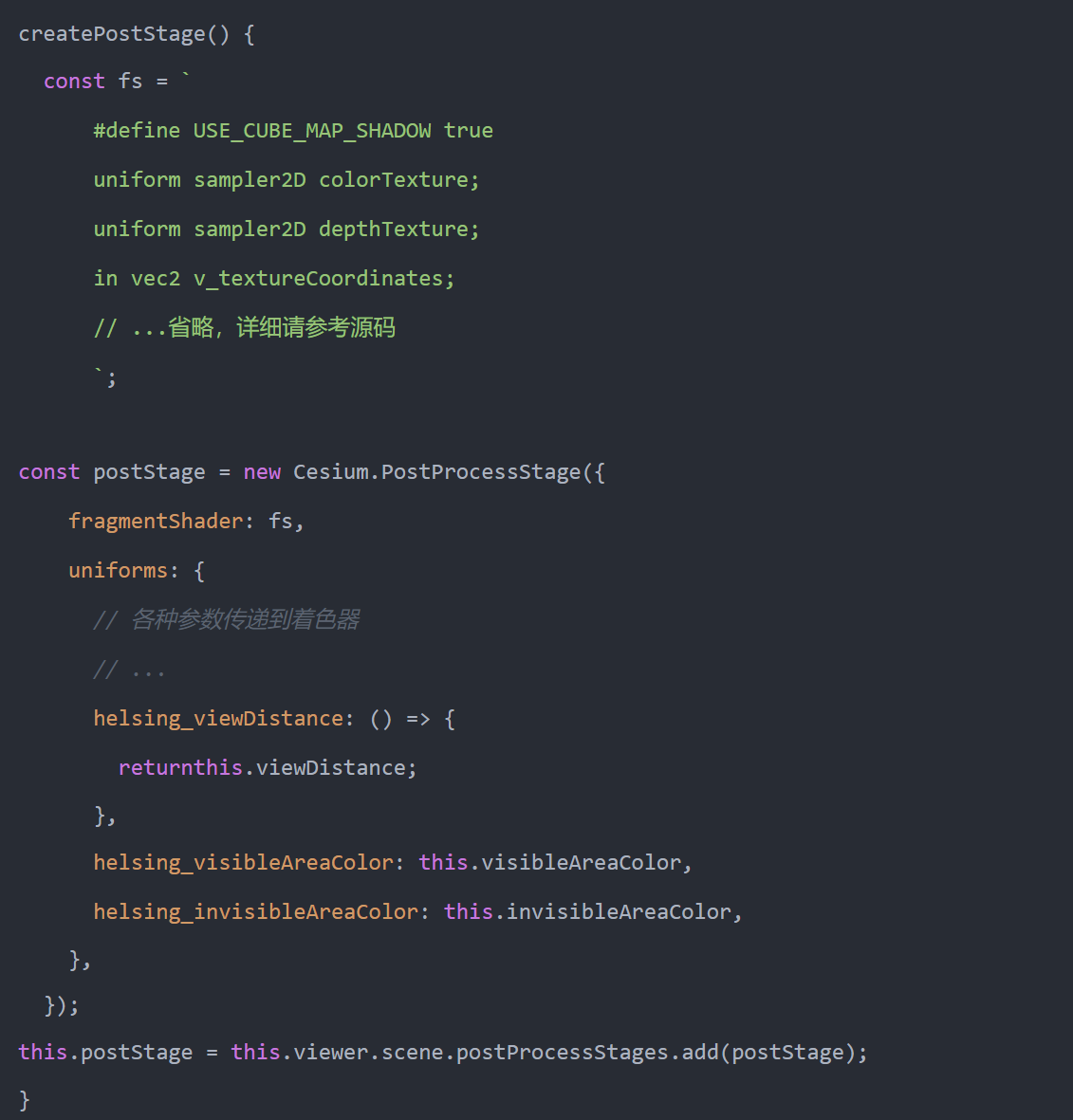
**3、创建阴影贴图，代码如下**



这个方法创建了一个ShadowMap对象，用于从光源（观测点）角度生成场景的深度信息。设置isPointLight为true表示从点光源出发，可以向各个方向投射视线。

4、创建后处理

后处理是可视域分析中最关键的部分，注意：着色器需要根据Cesium版本进行语法变更，webgl1跟webgl2的语法略有不同。 通过自定义着色器来渲染可视域分析的结果；



这个方法创建了一个后处理，其中包含自定义的片元着色器**。着色器通过比较每个像素点到观测点的距离与深度图中记录的距离，判断该点是否在视线范围内，然后对可见区域和不可见区域分别着色**。

**4、色器代码的工作流程是：**

获取当前像素的颜色和深度信息

转换到相机坐标系和世界坐标系

转换到观测点（虚拟相机）坐标系

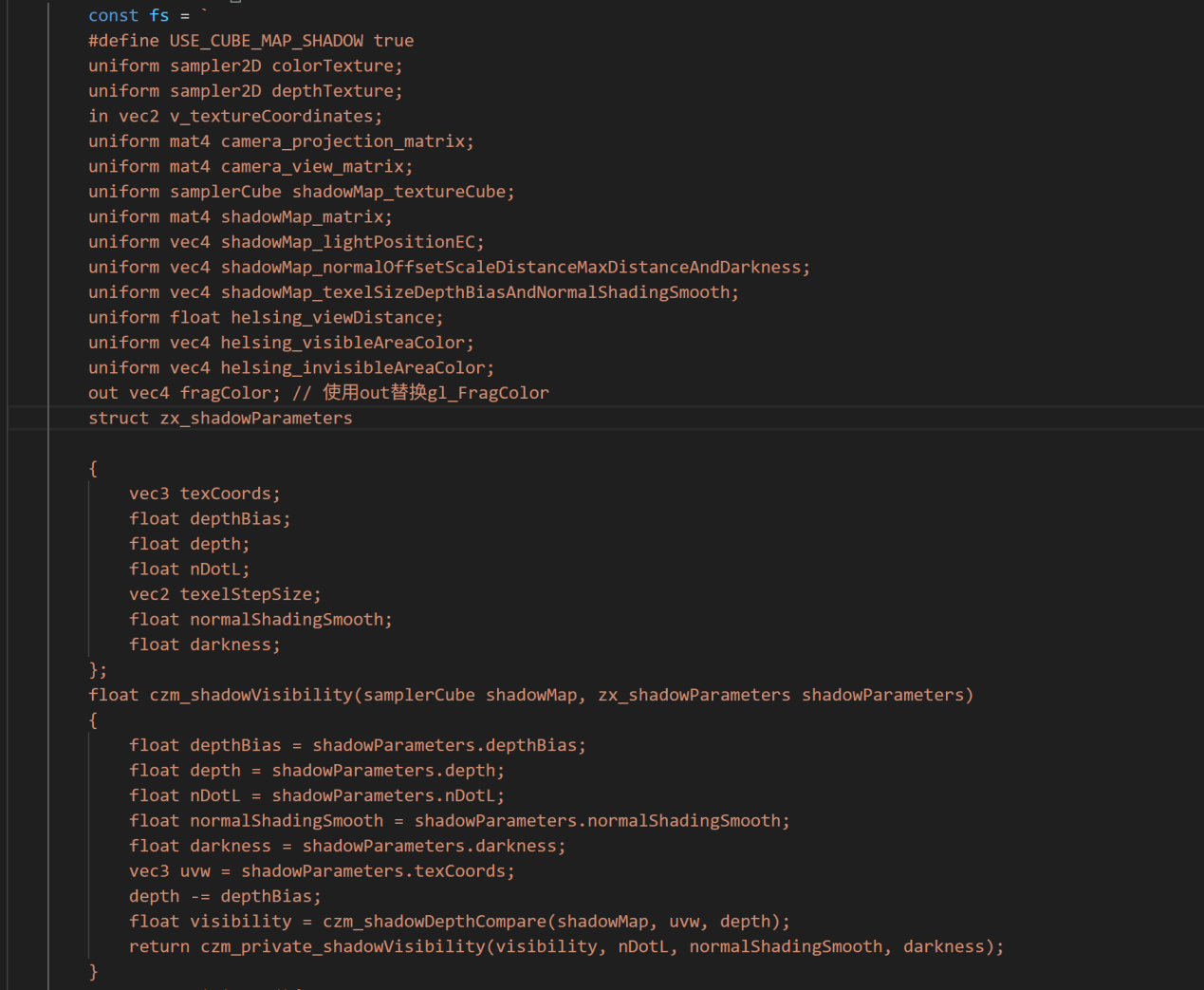
判断点是否在观测距离范围内

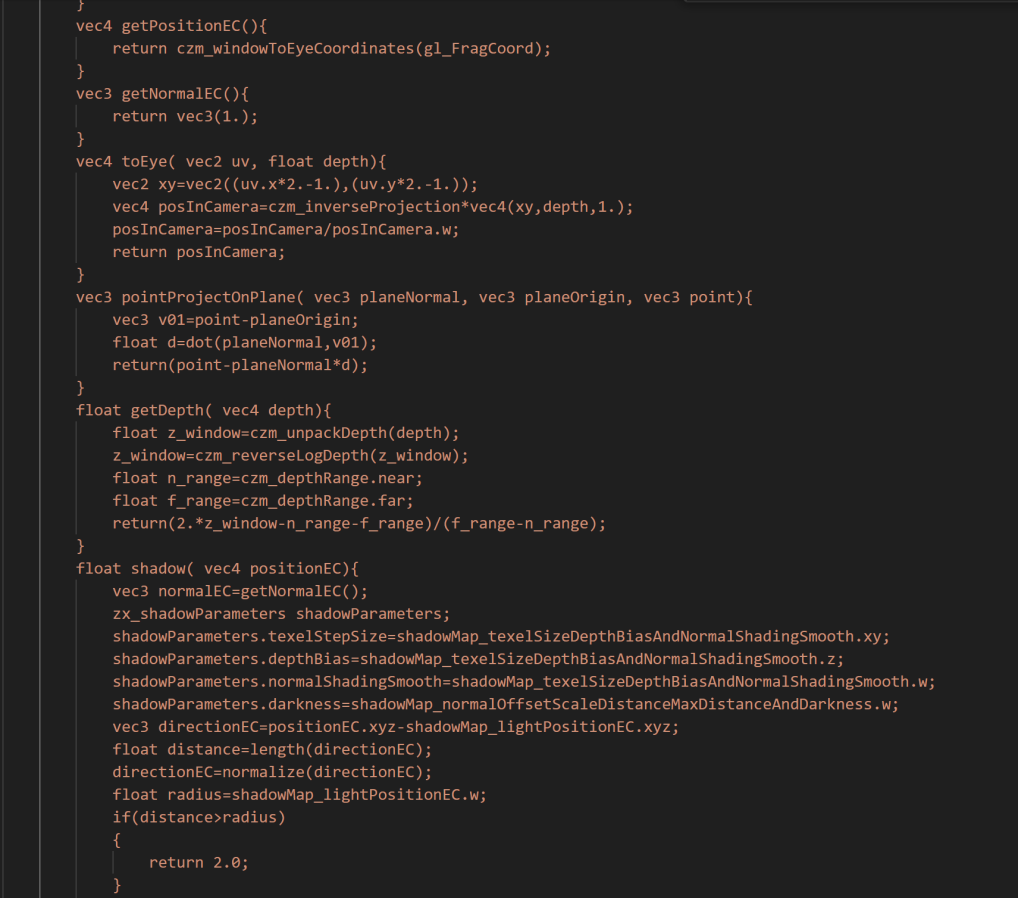
判断点是否在视锥体内

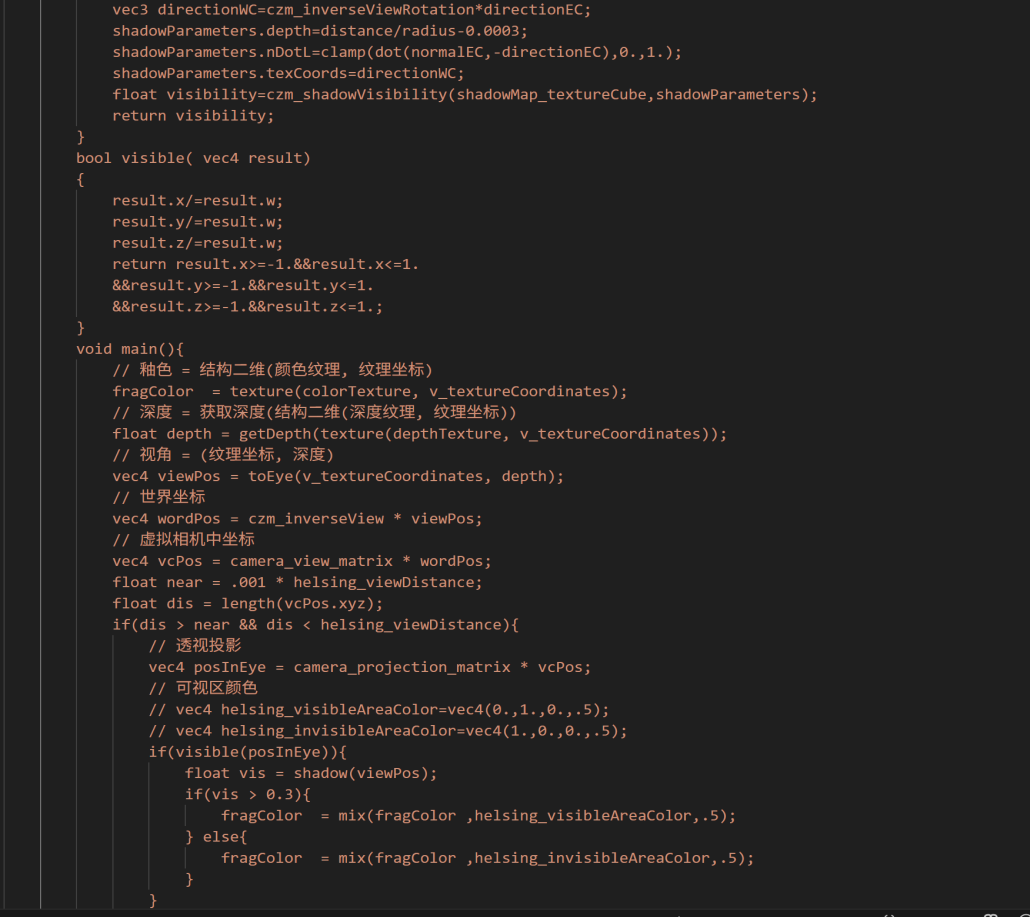
通过shadow函数判断点是否可见

根据可见性结果进行着色

代码如下：

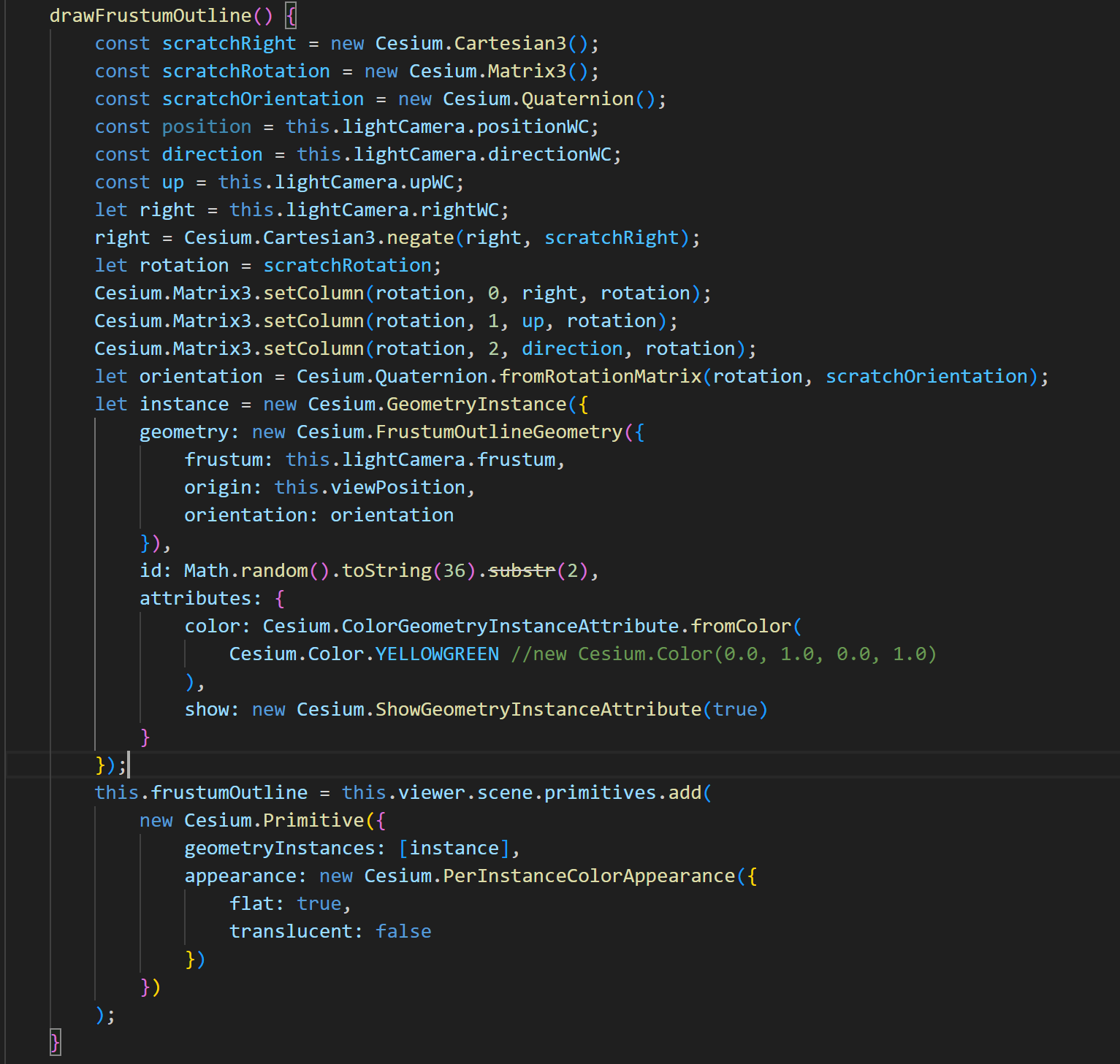




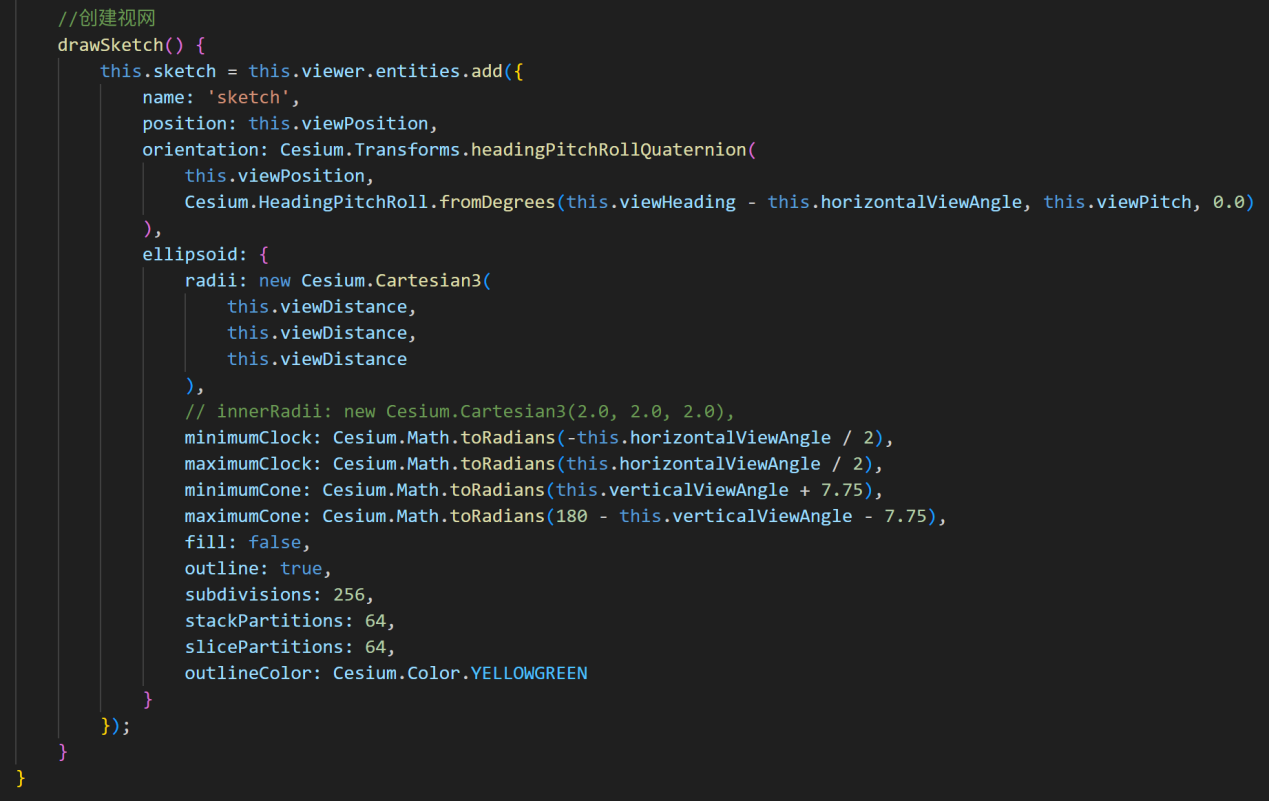


**5、创建视锥体和视网**

**创建视锥体：**



**创建视网：**



**注意：如果直接使用内置的地形，会出现都是可见区域这样的错误。**

**主要是因为**：Cesium 的默认地形（Cesium World Terrain）是基于 GPU 加速的地形渲染技术（如 GlobeSurfaceTileProvider），不是普通的 Primitive 或 Model。

而 ShadowMap 的原理依赖的是 从光源（或观察点）渲染视角下场景中所有参与阴影投影的图元（geometry）。

**解决方案：**

**方法 1：使用可投影的“实体地形”**

如果你只需要在一个小区域做精确分析：

使用自定义的地形 mesh，比如：

通过 3D Tiles 加载地形模型。

把 DEM 数据转成 .glb / .gltf 模型加载进场景。

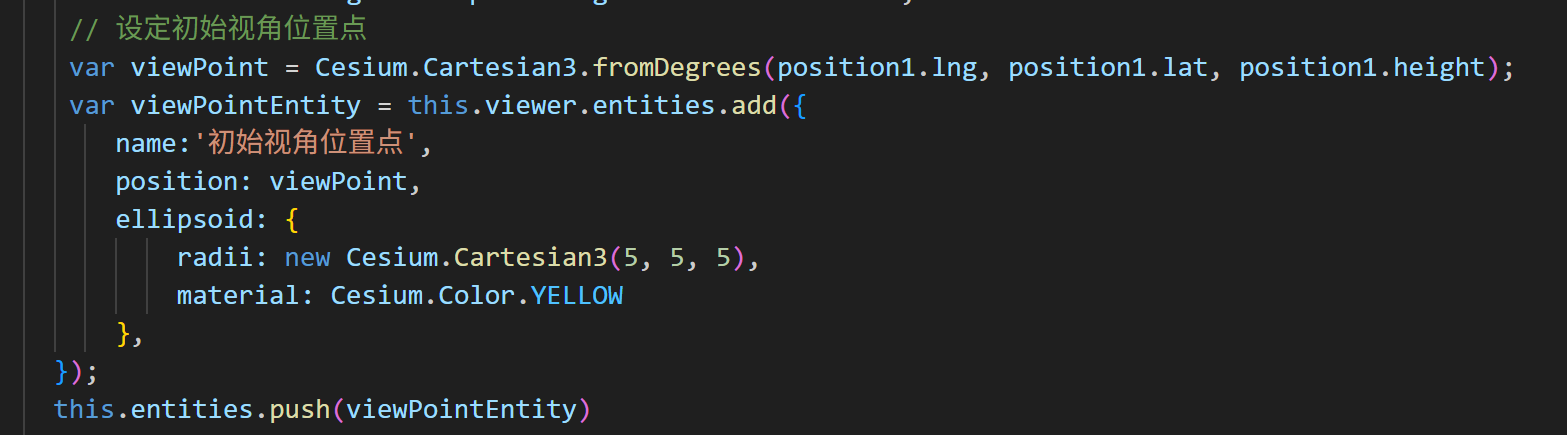
使用 Primitive 加载高程面片（可接受阴影）。

这样它们就能被 ShadowMap 采样，从而参与可见性判断

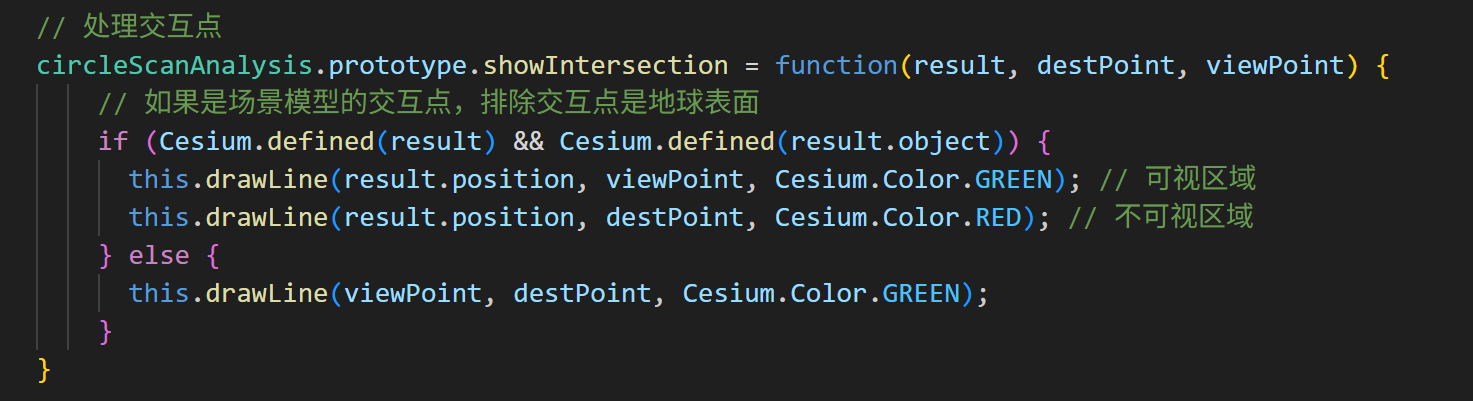
**方法 2：高级做法：自定义 Shader / 自定义 Ray Casting**

**核心代码如下：**

设定初始视角位置点



地形采样，计算和地形的交互点



绘制通视线，可见的为绿色，不可见的为红色。

