

基于 Three.js 的矿山隧道 3D 可视化模型设计与实现

刘仕琦 丁鹏 杨堃 陈旭东 徐佩尧

中冶赛迪信息技术(重庆)有限公司

DOI:10.12238/acair.v2i4.10324

[摘要] 随着矿山隧道工程的复杂性和规模的增加,传统的二维平面设计已无法满足现代化建设和管理需求。基于此,本研究介绍了一种基于Three.js的矿山隧道3D可视化模型设计与实现方法,该模型集成情报显示器、告警系统、安全出口标识等关键功能,实现对矿山隧道的实时监控和管理,以提高工程设计的效率和安全性。

[关键词] Three.js; 3D可视化; 几何模型; 材质; 点光源; 矿山隧道; 实时监控; 安全管理

中图分类号: TD5 **文献标识码:** A

Design and Implementation of 3D Visualization Model for Mining Tunnel Based on Three.js

Shiqi Liu Peng Ding Kun Yang Xudong Chen Peiyao Xu

Zhongye Saidi Chongqing Information Technology Co., Ltd.

[Abstract] With the increasing complexity and scale of mining tunnel engineering, traditional two-dimensional plane design can no longer meet the needs of modern construction and management. This study introduces a design and implementation method for a 3D visualization model of mine tunnels based on Three.js. The model integrates key functions such as intelligence display, alarm system, and safety exit signs to achieve real-time monitoring and management of mine tunnels, in order to improve the efficiency and safety of engineering design.

[Key words] Three.js; 3D visualization; Geometric model; texture of material; Point source; Mining tunnels; Real time monitoring; security management

引言

矿山隧道工程作为现代化建设的重要组成部分,其设计和管理的复杂性随着工程规模的扩大而日益增加。传统的二维设计方法已难以满足现代工程对于精确度和实时监控的需求。三维可视化技术以其独特的直观性和互动性,为隧道工程的现代化管理提供了新的视角和解决方案。本研究聚焦于矿山隧道3D可视化模型的设计与实现,旨在通过三维空间的构建,提供一个更为全面和深入的工程监控和管理工具。该模型不仅能够展示隧道的三维结构,还能够集成关键的工程功能,如照明、通风、交通指示等,实现对隧道环境的实时监控和管理。本研究不仅为矿山隧道工程的设计和管理提供了新的工具,也为三维可视化技术在工程领域的应用开辟了新视角,推动了工程可视化技术的发展,为工程安全和管理提供了新的思路。

1 相关技术概述

Three.js是一个基于WebGL的开源JavaScript库,它为3D图形的在线展示提供了强大的支持。它的核心优势在于能够简化3D图形的创建过程,允许开发者利用简单的代码实现复杂的3D场景。Three.js基本要素主要为:

(1) 场景: 场景是Three.js中的基本构建块,所有的3D对象、灯光和相机都被添加到场景中。

(2) 相机: 相机在Three.js中定义了观察者的视野和视角。

(3) 网格: 网格是由几何体和材质组合而成的3D对象。分别定义了物体的形状和物体的表面属性(如颜色、纹理和光泽度)。

(4) 渲染器: 渲染器负责将场景和相机结合,生成最终的图像并显示在屏幕上。

(5) 控制器: 控制器用于处理用户输入,以便在场景中移动相机或与3D对象交互。

2 模型设计与功能实现

2.1 矿山隧道3D可视化模型设计

2.1.1 隧道结构设计

隧道主体由5个部分组成: 内壁、外壁、底面、前截面、后截面,将5个几何体通过THREE.mergeGeometries类合并,并为其添加材质,设置材质颜色和透明度,方便观察。

(1) 使用THREE.CylinderGeometry类,通过配置底部外半径、顶部外半径、高度、圆柱体圆周分段数、顶部和底部的圆的分段式、封闭柱体两端、圆柱起始角度、圆柱结束角度创建

外壁、内壁半圆柱几何体。

(2) 使用THREE.Shape类, 定义底面二维轮廓, 使用其类方法moveTo(x, y) 移动到指定的点(x, y), 开始新的子路径, 类方法lineTo(x, y) 从当前点绘制一条直线到指定的点(x, y), 以此类推创建二维底面。使用THREE.ShapeGeometry类将THREE.Shape实例转换为三维几何体。

(3) 使用THREE.Shape类创建前后截面, 与底面不同, 前后截面需要使用类方法absarc(), 通过绝对坐标来定义圆弧的起始点、半径、起始角度和结束角度绘制弧形作为截面的外弧。

使用THREE.Path()类创建内弧路径, 外弧和内弧的起始角度和结束角度并不相同, 内弧需要一个偏移量angleOffset。

将内圆孔路径添加到shape实例的holes属性中, 用于THREE.ShapeGeometry类在创建几何体时, 会将内圆孔作为一个孔洞处理。

使用THREE.mergeGeometries类, 将多个几何体合并成一个几何体, 以减少绘制调用, 从而提高性能。

使用THREE.MeshStandardMaterial类创建外部灰色、双面渲染、透明度为0.5的材质, 并使用THREE.Mesh类将材质与几何体组合。

2.1.2 照明系统设计

创建环境光和一个光源组, 为光源组添加多个点光源和金属材料灯罩, 模拟隧道内部照明设备效果, 并在轨道控制器滑动轨道时体现环境明亮变化。

使用THREE.Group类创建一个lightGroup组对象, 作为将多个对象(如网格、光源等)组合在一起的容器, 便于统一管理和变换, 使用scene.add(lightGroup)方法将该灯光组添加到场景中, 使其包含的所有对象都能被渲染。

(1) 设置点光源之间的距离, 并计算需要的点光源数量numLights, 使用THREE.PointLight类循环创建点光源。

(2) 在每个点光源周围使用THREE.TorusGeometry(0.25, 0.05, 16, 100)类创建环的半径0.25、管道半径0.05、横向分段数16和纵向的分段数100的圆环几何体。

使用THREE.MeshStandardMaterial({ color: 0xc0c0c0, metalness: 1, roughness: 0.5 })类创建金属材质的圆环模拟灯罩, metalness参数用于配置材质金属质感, roughness参数用于配置粗糙程度。

通过THREE.Mesh类将几何体和材质组合成一个网格对象并添加到灯光组中, 使光源和其装饰物可以一起移动、旋转或缩放, 以此简化场景的管理和渲染。

2.1.3 风机设计

使用几何体组成风机的电机、扇叶、支架、遮罩, 并设置旋转动画。通过自定义创建方法创建风机扇叶, 循环三次调用创建间隔120°, 倾斜角度30°的扇叶。通过Three.js内置动画循环方法requestAnimationFrame(), 使扇叶组每次动画更新时绕z轴旋转0.03弧度。创建遮罩和支架几何体, 与风机扇叶、电机几何体合并, 调整位置构成整体风机。

使用THREE.Shape类中shape.arc()方法绘制扇叶角度为60度二维形状; 使用THREE.ExtrudeGeometry(), 将二维扇叶拉伸为三维几何体; 使用THREE.CylinderGeometry类创建电机, 并使用requestAnimationFrame()设置旋转动画, 变量isFanOn是一个布尔值, 用于控制风机是否开启, 叶片会以每次0.03弧度的速度旋转。

使用THREE.CylinderGeometry类和THREE.MeshBasicMaterial创建风机保护壳与支架的几何体与材质, 将几何体与材质通过THREE.Mesh类组合成网格, 并添加至风机几何体组。

2.1.4 情报显示器设计

情报板显示器由屏幕、支架、底座、滚动字幕组成。创建canvas画布对象, 设置画笔大小、字体、颜色, 使用THREE.CanvasTexture类创建Canvas纹理, 将画布添加至几何体组, 通过requestAnimationFrame()动画循环方法, 将文本的x轴位置减少1, 实现在水平方向上的滚动效果。

(1) 使用THREE.BoxGeometry(3, 1, 0.1)类创建宽度为3, 高度为1, 厚度为0.1的长方体几何体作为屏幕主体, 并为其创建材质, 将几何体与材质通过THREE.Mesh类组合成网格, 并添加至情报板几何体组。

(2) 使用THREE.CylinderGeometry类和THREE.MeshBasicMaterial创建底座与支架的几何体与材质, 将几何体与材质通过THREE.Mesh类组合成网格, 并添加至情报板几何体组。

(3) 创建一个HTML<canvas>元素, 并设置其宽度和高度, 获取画布的上下文(context), 用于在画布上绘制内容, 将画布背景填充为黑色, 并设置字体颜色为白色。

使用THREE.CanvasTexture(canvas)类创建Canvas纹理贴图, 创建自定义方法updateCanvas()用于更新画布内容。每次更新会清空画布, 将画布填充为黑色, 重新绘制文本内容通过变量textX--1控制字幕滚动的速度, 如果字幕完全滚出视野, 重置textX, 使字幕从右侧重新出现, 并通知THREE.js更新纹理, 以显示最新的画布内容。

2.1.5 交通指示灯设计

交通指示灯由容器、红灯、绿灯、支架组成。创建容器、红灯、绿灯、支架几何体组, 并为每个几何体设置标识, 以便在控制面板事件触发时, 正确识别需要变化的几何体。

(1) 通过THREE.BoxGeometry(1, 2.5, 1)类创建宽度为1, 高度为2.5, 厚度为1的立方体几何体作为交通指示灯主体, 并为其创建材质, 将几何体与材质通过THREE.Mesh类组合成网格, 将该几何体命名为“container”并添加至情报板几何体组。

(2) 使用THREE.CylinderGeometry类和THREE.MeshBasicMaterial分别创建红灯、绿灯、支架的几何体与材质, 将几何体与材质通过THREE.Mesh类组合成网格, 分别命名为“RedLight”、“greenLight”、“stand”并添加至指示灯几何体组。

2.1.6 告警系统设计

告警系统利用隧道内点光源颜色交替切换实现告警效果, 并控制面板在触发警报时开启警报音效, 创建自定义开启、关闭

警报方法,将隧道点光源组光源颜色属性重置。

定义布尔类型变量toggle,开启告警系统时,执行StarPlay Alarm Audio(true)方法,播放警报音效,在循环体setInterval()方法中,每0.5秒改变一次toggle值,将点光源组中的点光源颜色在红色白色之间交替更换。

关闭告警系统时,执行StarPlayAlarm Audio(false)方法,关闭警报音效,使用点光源color属性的set方法,将点光源颜色重置为白色,并关闭开启告警系统方法中的循环体,避免内存泄露。

2.1.7 监控设备设计

使用THREE.CylinderGeometry类和THREE.MeshBasicMaterial分别创建监控设备主体、镜头、镜头玻璃、支架、底座的几何体与材质,将几何体与材质通过THREE.Mesh类组合成网格,使用CylinderGeometry方法创建。

2.1.8 紧急逃生出口标识设计

紧急逃生出口标识由边框、canvas绘制图案和点光源组成。使用THREE.BoxGeometry(0.5,0.25,0.01)类创建宽度为0.5,高度为0.25,厚度为0.01的长方体几何体作为出口标识框架,并为其创建金属质感材质,将几何体与材质通过THREE.Mesh类组合成网格,并添加至情报板几何体组。使用THREE.CanvasTexture类创建canvas纹理,也可通过纹理贴图代替,canvas画布绘制紧急逃生出口内容,并添加canvas材质。使用THREE.PlaneGeometry(0.5,0.25)创建一个宽为0.5,高为0.25的平面几何体。创建两个THREE.Mesh实例:signMesh1和signMesh2,作为标识正面和反面(旋转180度),并将它们添加到场景中,在标识中间添加点光源。

3 模型控制面板设计

控制面板事件与模型绑定,可控制交通指示灯红绿灯光开启与关闭,控制风扇的转动与停止,控制情报显示器的文字内容,控制警报系统开启与关闭。

3.1 控制信号灯

从信号灯几何体组trafficLightGroup中遍历材质实体,寻找

名称为红灯、绿灯的实体,通过事件触发时trafficLightColor变量改变控制实体显隐。

3.2 控制风机转动

控制面板事件触发,isFanOn变量改变,创建风机的方法中根据isFanOn变量值控制风机转动。如:consttoggleFan={()=>{isFanOn.value=!isFanOn.value}}

3.3 控制告警系统

控制面板事件触发告警系统开启或关闭方法。

```
consttoggleAlarm={()=>{isAlarmActive.value?deactivateAlarm():activateAlarm()}}
```

3.4 控制情报显示器内容

控制面板事件触发,updateInfoBoard方法,将情报显示器几何体组中textContent变量值和滚动位置重置为变化值。

4 总结

本文通过使用Three.js框架,详细介绍了矿山隧道3D可视化模型的设计与实现。通过创建情报板、报警系统和安全出口标识等功能模块,实现了矿山隧道的实时监控和管理。该模型为矿山隧道的设计和管理提供了直观、实时的可视化工具,提升了工程管理的效率和安全性。未来可以进一步优化模型的细节和功能,实现更加全面的矿山隧道管理系统。

[参考文献]

[1]林杰,祝江林.基于Three.js引擎模块的隧道三维可视化系统设计[与实现[D].武汉:武汉理工大学,2023.

[2]叶凤华.3D动画场景光照渲染算法的效果优化[D].广东:广东工业大学,2015.

[3]Three.js官方文档[EB/OL]https://threejs.org/.

[4]柳杉.使用three.js搭建3d隧道监测[EB/OL]https://juejin.cn/post/7273987266523136056.

作者简介:

刘仕琦(1994--),男,汉族,重庆合川人,本科,中冶赛迪信息技术(重庆)有限公司,研究方向:Three.js;3D可视化;几何模型;矿山隧道。