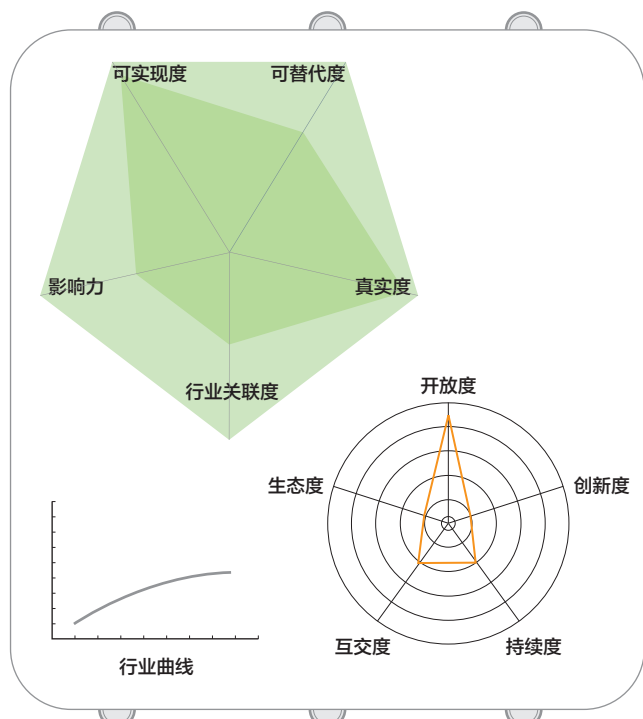


# 基于 WebGL 的地下管网三维可视化及应用

张旭敏 彭 靖



张旭敏 彭 靖

华北地质勘查局五一九大队

张旭敏（1989—），女，硕士研究生，工程师，软件开发人员，研究方向：三维 WebGIS。



城市地下管网是指城市范围内供水、排水、燃气、热力、电力、通信、广播电视、工业等管线及其附属设施，各类管线在地下空间交错纵横，保障了城市发展的正常运行，是保障城市运行的重要基础设施和“生命线”。由于地下管网的重要性及其错综复杂的空间分布情况，准确直观地掌握管网的分布现状，为城市规划管理提供重要信息极为重要，而且还可以有效避免城市安全事故的发生。

随着城市信息化进程的加快及云计算、人工智能等技术的快速发展，城市建设管理步入了数字化智能化时代。由于地下管网的重要性和空间分布的复杂性，地下管网的信息化管理的需求尤为迫切。目前地下管网的信息化建设以二维方式为主，但二维图形方式难以直观反映各类管线间的垂直空间分布关系。近些年“智慧城市”“数字孪生”等概念席卷全球，三维可视化技术逐渐成熟，三维模型方式较二维图形方式在表现管网复杂的空间分布及逼真程度等方面展现出较为明显的优势，能更好地满足地下管网信息化管理的需求，因此地下管网的三维化是数字化城市发展管理的必然趋势。

## 基于 WebGL 的三维可视化的相关技术

### HTML5 技术

HTML5 是 HTML（超文本标记语言）的第 5.0 版本，主要用于构建及呈现互联网内容，是互联网开发的核心技术之一。HTML5 实现了对 HTML4 版本的革新，为 Web 开发带来了许多新特性和新功能。主要的特性如下所示。

**语义化标签：**HTML5 引入了一系列新的语义化标签，如 <header><footer><nav> 等，这些标签使得网页结构更加清晰，有利于搜索引擎优化（SEO）和无障碍访问。

**多媒体支持：**HTML5 提供了内置的多媒体支持，包括 <audio> 和 <video> 标签，使得网页可以直接播放音频和视频内容，不再需要依赖第三方插件如 Flash。

**Canvas 绘图：**HTML5 的 <canvas> 标签允许开发者使用 JavaScript 在网页上绘制图形、动画和图像，为 Web 应用添加了更多的交互性和视觉效果。

**地理定位：**HTML5 通过 navigator.geolocation API 提供了一种在 Web 应用程序中获取用户位置的标准方法，使得开发者可以创建基于位置的应用程序。

### JavaScript 脚本语言

JavaScript 是一种轻量级的解释型或即时编译型的编

程语言，它常与 HTML 和 CSS 一起使用，用于创建动态网页、Web 应用程序等，在 Web 开发中得到了广泛应用。JavaScript 代码支持跨平台运行，几乎可以运行在现有的所有浏览器上，无论是 Windows、macOS 还是 Linux 操作系统。

JSON 是 JavaScript 的子集，是一种轻量级的数据交换语言。它是一种数据格式，不是一种编程语言。JSON 具有易于编写、阅读、解析和生成等优点，因此广泛用于 Web 开发中进行数据的传输。服务器可以发送 JSON 数据给客户端，客户端利用 JavaScript 解析数据，并更新客户端内容。同样，客户端也可以发送 JSON 数据给服务器，服务器上程序解析这些数据并执行相应操作。GeoJSON 是基于 JSON 的地理空间信息数据交互语言，它专为存储网络地理信息数据设计，能够表示点、线、面等矢量数据。

### PostgreSQL 数据库

数据库是 Web 开发的核心组件，用于存储、检索、更新和管理 Web 开发所需的数据。PostgreSQL 是一个功能强大的开源对象关系数据库管理系统，能够在多种操作系统中运行并支持大部分 SQL 标准。其通过 PostGIS 扩展，实现了对地理空间数据的存储、检索、分析和处理。

### WebGL 技术

WebGL (Web Graphics Library) 是一种在 Web 浏览器中渲染 3D 图形的 API，它基于 OpenGL ES，允许开发者使用 JavaScript 和 HTML5 的 <canvas> 元素来创建和显示 3D 图形。传统的 Web3D 应用需要安装相应的插件才可在 Web 浏览器端进行 3D 图形渲染，不同的浏览器和操作系统对插件版本的需求不同，插件维护困难使用不便。而 WebGL 是一套跨平台的 JavaScript API，它支持浏览器内嵌使用，用户无需安装任何插件或软件，即可在现代浏览器中访问 WebGL 应用程序。它可以在支持 WebGL 的各种平台上运行，包括 Windows、MacOS、Linux、Android 和 iOS 等。而且它通过利用 GPU 的并行计算能力，实现高性能的 3D 图形渲染，比传统的 2D 渲染具有更快的速度和更好的图像质量。

### 基于 WebGL 的三维 GIS 开源平台

Cesium 是一个开源的基于 WebGL 实现的 JavaScript 三维地球框架。Cesium 不仅支持 Web 浏览器无插件创建 3D、2.5D、2D 地图而且还可以利用 WebGL 来实现图形硬件加速渲染、跨平台和真实动态数据可视化图形渲染。主要特点如下所示。

全球高精度地形数据可视化；

支持多种资源的影像图层加载，包括 WMS，TMS，WMTS 以及时序影像；

支持 KML、GeoJSON、TopoJSON 等标准矢量格式的矢量数据加载，并可可视化点、线、面、体等各种几何体；

支持 GlTF 和 3DTiles 格式的三维模型数据加载。

## 基于 WebGL 的地下管网三维可视化方法

### 技术路线

国内外地下管线三维模型的建立主要是依靠专业的三维建模软件 (3DsMax、Blender、SketchUp 等) 按照地下管线普查数据手工构建，但地下管线绵延数百里，管线数量多且尺寸类型不一样，运用专业三维建模软件构建三维管线模型会耗费大量人力和时间。现代的三维管线建模技术如利用近景摄影测量技术三维建模和利用三维激光扫描技术建模等主要对地上管线的三维建模比较有效，不适用于地下管线的三维模型构建，而且这些方法建立的三维管线模型无法实现管线的分类管理和属性管理。而目前针对三维管线客户化领域主要基于桌面 GIS 应用，技术已经比较成熟，但是基于 Web 的三维管线可视化多数需要安装浏览器插件，无法满足用户对高效、高扩展、轻量化应用的要求。因此本文通过研究 WebGL 技术、空间地理数据库、三维模型可视化的基础上，设计并提出了一种基于 WebGL 的三维地下管线自动渲染方法。根据地下管网数据的特点，对地下管网数据进行抽象和转换，实现管点、管线的标准化三维建模渲染。

本方法对地下管网数据库采用统一的数据存储模型，按照空间数据和属性数据一体化存储的方式，对地下管网数据进行组织和分类。通过对管网数据进行自动投影转换，提供地理坐标系下基于 GeoJSON 的管线数据检索方法；建立管网数据标准构建模型库；依据管网数据检索结果，基于地理空间位置自动计算管网模型矩阵，结合 WebGL 批量渲染技术，实现管网三维模型的自动化构建；对管网部件中的弯头、三通等管件，研究形成了一套模型矩阵和弯头角度计算算法；结合 JavaEE 后台服务接口，通过模型 ID 与接口匹配，实现全要素管件识别方法。

该方法包括的步骤如下所示 (图 1)。

1. 地下管线数据整理：其中地下管线数据指管线普查获取到的二维标准的各种类型的管线管点数据；并对上述数据进行标准化处理得到统一标准的地下管线数据；
2. 地下管线数据入库：将标准化后的地下管线数据的空间及属性信息录入到空间地理数据库中；
3. 投影变换：基于空间地理数据库将入库的地下管线数据从投影坐标转换为地理坐标；
4. 三维模型库构建：根据管线管点类别分别构建管线、管点三维模型库；
5. 管件模型矩阵计算：利用管线起终点计算得到三维管线模型的模型矩阵；利用管点坐标与管点连接的管线起终点计算得到三维管点模型的模型矩阵，并根据上述参数进一步计算三维管点模型各部件的平移旋转缩放矩阵；

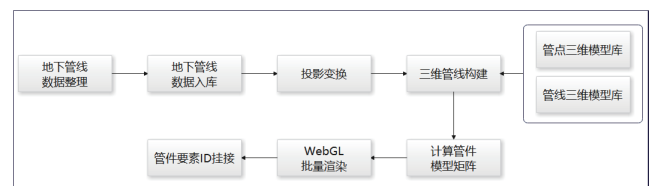


图 1 基于 WebGL 的地下管网三维可视化流程图

6.WebGL 自动化渲染：利用基于 WebGL 技术开发的 Cesium 三维地球引擎，对上面步骤的管件三维模型按照模型矩阵进行自动化渲染，在地图上构建出三维地下管线场景；

7. 管件要素 ID 挂接：结合 JavaEE 后台服务接口，通过模型 ID 与接口匹配，实现全要素管件识别。

基于 WebGL 的地下管网三维可视化方法利用 WebGL 技术结合 PostGIS 空间地理数据库，可以快速地在浏览器端对地下管线数据进行三维实时渲染、管线数据实时联动更新渲染等，大大节省了用于人为建模的人力和时间，同时满足了三维管线系统高效、高扩展、轻量化应用的要求，为城市三维管网信息化、地下基础设施管理等领域具有重要的借鉴价值。

### 管段三维可视化

管段模型是地下管网三维可视化过程中最主要且数据量最大的三维模型对象，三维管段模型根据二维管段数据的空间位置和走向进行绘制，管段模型可以重复利用但其数据量较大，如果直接利用 Cesium 引擎中逐个绘制几何实体的方法会造成浏览器绘制压力过大导致系统崩溃。因此本文采用 Cesium 中支持批量加载大量 gltf 格式三维模型的 ModelInstanceCollection 接口实现海量管段模型加载，具体实现方式如下。

1) 管段模型准备：从管段三维模型库中获取所需的 gltf 或 glb 格式的管段模型，可通过使用建模软件（3DMax 或 Blender 等）制作标准规格（长度为 1 米直径为 1 米或长宽高均为 1 米）的三维管段模型，然后输出为 gltf 或 glb 格式。

2) 管段模型矩阵计算：首先获取管网数据库中的管段的起终点坐标分别为  $(x_1, y_1, z_1)$ 、 $(x_2, y_2, z_2)$ ，管线单位方向向量为  $(\frac{x_2-x_1}{|x_2-x_1|}, \frac{y_2-y_1}{|y_2-y_1|}, \frac{z_2-z_1}{|z_2-z_1|})$ 。在 Cesium 中定义欧拉角（HeadingPitchRoll）来表示模型旋转的信息，Heading 是方向角或航向角，表示旋转模型绕模型自身 Z 轴旋转且逆时针旋转为正；Pitch 是俯仰角，表示旋转模型绕模型自身 Y 轴旋转，逆时针旋转为正；Roll 是翻滚角，表示旋转模型绕模型自身 X 轴旋转，顺时针旋转为正，具体如图 2。

在计算得到：

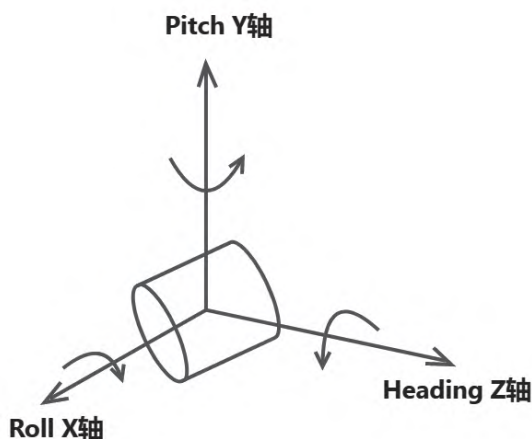


图 2 欧拉角示意图

$$heading = -(\frac{x_2-x_1}{|x_2-x_1|}, \frac{y_2-y_1}{|y_2-y_1|}, 0) \cdot (1, 0, 0) = -\frac{x_2-x_1}{|x_2-x_1|}$$

$$pitch = -(\frac{x_2-x_1}{|x_2-x_1|}, 0, \frac{z_2-z_1}{|z_2-z_1|}) \cdot (1, 0, 0) = -\frac{x_2-x_1}{|x_2-x_1|}$$

$$roll = 0$$

进而得到按 XYZ 坐标轴旋转矩阵为：

$$Rx(roll) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(roll) & -\sin(roll) & 0 \\ 0 & \sin(roll) & \cos(roll) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Ry(pitch) = \begin{pmatrix} \cos(pitch) & 0 & \sin(pitch) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(pitch) & 0 & \cos(pitch) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rz(heading) = \begin{pmatrix} \cos(heading) & -\sin(heading) & 0 & 0 \\ \sin(heading) & \cos(heading) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rxyz = Rx(roll) \cdot Ry(pitch) \cdot Rz(heading)$$

管段缩放矩阵：

$$Sx = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

$$Sy = ds$$

$$Sz = ds \quad ds \text{ 为管段的管径}$$

$$S = \begin{pmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

管段平移矩阵：

$$T = \begin{pmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

得到最终模型矩阵：

$$modelMatrix = Rxyz \cdot S \cdot T$$

大量 gltf 格式三维模型批量绘制：本文采用 ModelInstanceCollection 接口来批量加载海量 gltf 三维管段模型。该接口通过生成三维模型实例集合实现对同一模型多种形态的海量数据的加载优化，例如大范围的树木、间隔摆放的路灯以及本文的管段模型。在这个集合中所有实例都引用相同的基础模型，但又具有独特的实例属性，如模型矩阵、拾取 ID 等。每个模型实例是相对于中心渲染的。为了获得最佳效果，模型的位置应该彼此靠近。否则如果模型放置在地球的相对两侧，则可能会出现精度问题。具体加载代码为：

```
let instances = [];
instances.push({
  modelMatrix: modelMatrix,
  batchId: id
});
viewer.scene.primitives.add(
  new Cesium.ModelInstanceCollection({
    url: './管道.glb',
```



```
instances: instances,  
    })  
)
```

### 自适应管件三维可视化

三维管网中的管件包括了阀门、方井、圆井等普通管件，可以按照上述管段绘制方式进行绘制；对于弯头、三通、四通等自适应管件，需采用单独绘制的方式。以弯头为例，制作带有骨骼的弯头模型，骨骼位置以管件需要进一步旋转的位置进行确定，本文制作的弯头模型如图 3 所示。

带骨骼的三维弯头模型制作完成后，先根据弯头所在的空间位置和方向确定弯头的初始姿态，然后获取弯头连接的

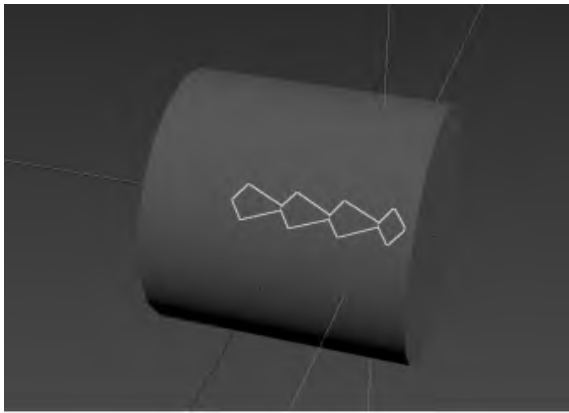


图 3 带骨骼的三维弯头模型

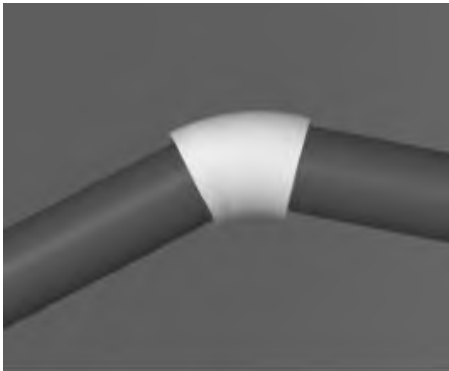


图 4 自适应弯头效果

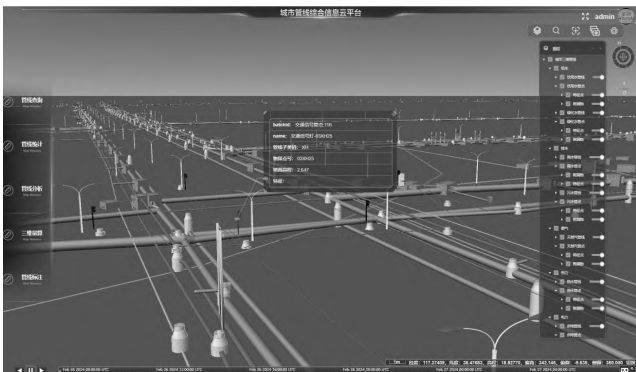


图 5 系统主界面

管段的起终点坐标，根据起终点位置和弯头位置调整弯头骨骼的角度，从而达到弯头随管段弯曲的状态。具体计算弯头骨骼的旋转角度方法是分别计算弯头连接的两个管段的欧拉角并进行相减得到骨骼的旋转角度，实现效果如下。

上述内容详细叙述了三维弯头的自适应绘制方法，同理三通、四通也可利用该方法进行渲染绘制，在绘制时需要考虑多个方向的骨骼旋转即可。

## 原型系统设计与应用

本文以 B/S 模式的若依框架为基础，结合 WebGL 图形硬件加速渲染技术，依托 Cesium 平台，设计实现了基于 WebGL 的城市管线综合信息平台。该平台实现了对地下管线数据的三维可视化，并在此及基础上开发了管线查询统计、管线分析、管线标注等功能，一定程度上提高了城市地下管线管理的水平和效率。

### 系统总体设计思路

城市管线综合信息平台基于 B/S 架构，采用分层模式开发。设计系统分为三层：底层为数据的预处理层，主要负责对二维管线矢量数据的入库操作；中间是服务层，主要管理数据的存储以及响应上层客户端的数据请求；上层客户端是直接面向用户的浏览器操作界面，主要完成数据请求，三维可视化及动态交互等具体应用。

### 系统功能设计

城市管线综合信息平台实现了二维管网数据在 WebGL 客户端的快速三维可视化及场景漫游，并允许用户与系统进行良好的交互，满足了城市地下管线专业技术人员对城市地下管线信息化管理的需求。目前系统的功能模块主要分为管线查询、管线统计、管线分析、三维量算、管线标注五个模块，每个模块下包含了相关的若干个具体功能。其中管线查询模块包括特征查询、附属物查询、管径查询、材质查询、条件查询、复合查询、缓冲区查询功能；管线统计模块包括特征分类统计、附属物分类统计、管径分类统计、材质分类统计、管点统计、管线统计功能；管线分析模块包括横断面分析、纵断面分析、净距分析、碰撞分析、设施搜索、覆土分析、流向分析、追踪分析、连通分析、爆管分析、地形开挖功能；三维量算模块包括距离量算和面积量算功能；管线标注模块包括管线标注、角度标注和扯旗标注功能。系统界面如图 5 所示。

## 总结

本文首先介绍了基于 WebGL 的三维可视化相关技术，然后研究了基于 WebGL 的地下管线三维可视化方法的技术路线及具体算法，最后使用 HTML5、JavaScript 和 WebGL 等技术，依托 Cesium 平台成功开发了城市管线综合信息平台，实现了地下管线数据的三维可视化及相关管线专题应用，为进一步提升三维管线信息化管理水平提供了技术支撑。