

· 信息技术 ·



基于 BIM+GIS 的航道三维漫游系统设计与实现

袁占全¹, 李明益²

(1. 中交第二航务工程勘察设计院有限公司, 湖北 武汉 430071;
2. 中交(苏州)城市开发建设有限公司, 江苏 苏州 215000)

摘要: 针对航道工程空间分布特性, 将广域航道工程划分成若干区域, 利用分区域的三维场景数据组织技术进行动态调度, 通过插件工具对 BIM 模型进行轻量化减面处理, 减少人工减面工作量, 并提出 Unity 加载 GIS 多源数据技术方法, 解决了航道工程大体量 BIM+GIS 三维数据在 Unity 引擎中无法加载、渲染卡顿、人工轻量化模型工作量大等项目实际应用过程中的难题。同时利用 Unity 引擎开发漫游系统, 使得超长航道 BIM+GIS 三维数据加载与高质量的漫游可视化效果兼得成为可能, 对 BIM 技术在航道工程中的推广应用具有重要价值。

关键词: 航道工程; BIM+GIS; Unity 引擎; 漫游系统

中图分类号: U 615

文献标志码: A

文章编号: 1002-4972(2022)02-0153-05

Design and implementation of three-dimensional waterway roaming system based on BIM + GIS

YUAN Zhan-quan¹, LI Ming-yi²

(1.CCCC Second Harbor Consultants Co., Ltd., Wuhan 430071, China;
2.CCCC(Suzhou) Urban Development and Construction Co., Ltd., Suzhou 215000, China)

Abstract: According to the spatial distribution characteristics of waterway engineering, wide-area waterway engineering is divided into several regions, and the dynamic scheduling is carried out by the data organization technology of three-dimensional scenes in different regions. Plug-in tools are employed for the surface reduction of building information modeling (BIM), which ease the workload of manual surface reduction. In addition, the technology of loading geographical information system (GIS) multi-source data by Unity is proposed to solve the problems in practical project applications, such as loading failure of large-scale three-dimensional BIM + GIS data of waterway engineering in Unity, sluggish rendering, and huge workload of making model lightweight manually. In addition, a roaming system is developed utilizing the Unity engine, which enables the loading of three-dimensional BIM + GIS data of super-long waterway and high-quality roaming visualization. The system is of great value to the promotion and application of BIM technology in waterway engineering.

Keywords: waterway engineering; BIM + GIS; Unity engine; roaming system

近年随着计算机三维渲染技术的发展, 对三维场景实时渲染和可视化漫游技术的应用需求日益增加, 特别是在工程建筑行业, 因为 BIM 技术发展, BIM 成果展现要求三维渲染的效果更加逼真、精美。目前市场主流解决方案是采用三维 GIS

平台与 Unity 3D 两种方式搭建三维漫游系统, 但存在多源数据、大数据量与场景渲染效果不能兼得问题^[1]。三维 GIS 平台可以承载大数据量 BIM+GIS 模型, 但其渲染效果不够逼真精美; 游戏引擎 Unity 3D 虽然三维渲染效果炫酷, 但无法

承载多源 GIS 数据、大面积地形和影像、精细 BIM 模型数据。而航道工程分布广, 对大体量数据加载、三维水面渲染与漫游效果都有非常高的要求。为此, 笔者提出基于 BIM+GIS 的航道三维漫游系统, 重点解决航道工程大体量多源三维数据、三维场景渲染与漫游效果不能兼得的问题, 助力 BIM+GIS 三维技术在航道工程行业的应用。

1 研究思路

依托长江干线武汉—安庆段航道 6 m 水深航道整治工程项目, 根据航道工程空间分布特征和航道 BIM 模型特点, 基于 Unity 3D 开发 BIM+GIS 的航道三维漫游系统, 解决航道工程大体量三维数据、三维场景渲染与漫游效果不能兼得问题。技术路线见图 1, 研究内容主要包括:

- 1) 解决 Unity 加载 GIS 数据技术问题, 同时采用自动化插件对 BIM 模型进行轻量化处理, 减少人工减面的工作量;
- 2) 根据航道工程空间分布特征提出“分区动态数据组织管理”+LOD 三维动态渲染相结合技术;
- 3) 基于 Unity 3D 游戏引擎设计实现手动漫游控制和自动飞行漫游功能。

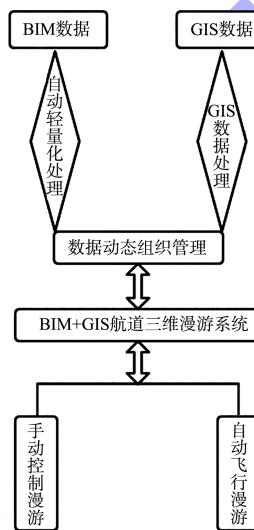


图 1 技术路线

2 BIM+GIS 三维航道数据处理

2.1 BIM 模型轻量化技术

用 BIM 模型详尽描述一栋建筑物或者构筑物的外在和内在构造将涉及大量的建筑细节, 所以

一般情况下 BIM 模型的体量较大^[2-3]。模型轻量化的核心思想在于尽量保持模型原状态下减少构成物体的三角形的数量。

轻量化传统技术路线是在 3DMAX 软件中进行人工减面处理, 其工作量非常大, 消耗大量人力物力。Unity 3D 的 MeshSimplify 插件中包含多种轻量化的算法, 其核心算法也是平面拟合原理, 最大特点是可以遍历父物体下的所有子物体, 然后分析每个子物体的构成, 继而针对性地进行轻量化处理。MeshSimplify 插件是一款开源的插件, 如果某些算法不适用于当前的模型, 可以继承源码的类之后进行修改。一般情况下直接使用该插件即可, 在需要轻量化模型导入插件, 设置简化百分比即可, 效果如图 2 所示。

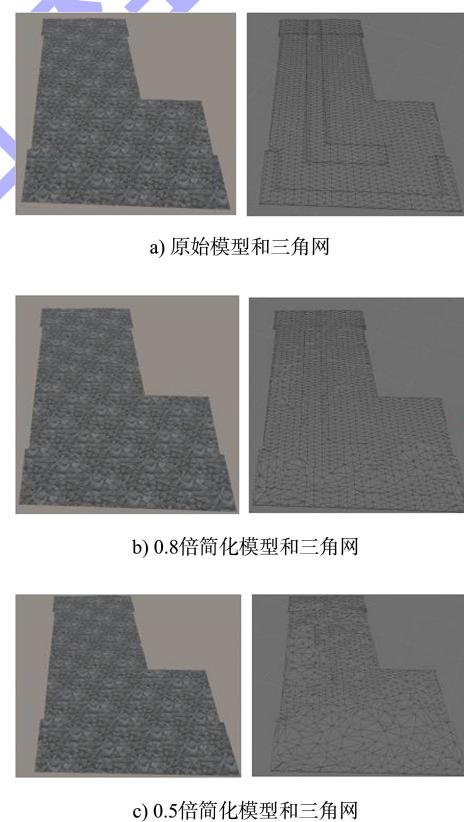


图 2 模型自动化处理对比

图 2a) 展示未进行轻量化的三角形构成及分布情况, 图 2b) 展示对其进行 0.8 倍轻量化后三角形的构成, 图 2c) 展示对其进行 0.5 倍量化后三角形的构成。通过这 3 组不同程度轻量化的模型对比可知, 在模型容量变小的同时, 航道 BIM 模型基本保持原模型的形态特征, 满足实际航道工程应

用的需要。

2.2 GIS 数据处理技术

2.2.1 加载栅格数据

1) 地形数据在 GlobalMapper 等 GIS 软件进行处理，并导出 DEM 文件；

2) DEM 导入 3DMAX 生成实体模型, 加载影像进行贴图, 同时优化数据纹理色彩等;

3) 将处理好的模型导出实体模型 .FBX，在 Unity 3D 中利用 MeshSimplify 插件对数据进行轻量化处理，然后加载到三维场景中。

加载栅格数据流程见图 3。

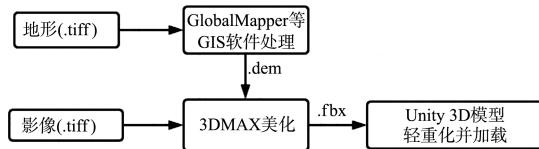


图 3 Unity 3D 加载栅格数据 (DEM、DOM)



图 4 长江武安段航道整治工程分布

3.2 航道工程分布特点

航道工程是开拓航道和改善航道航行条件的工程。项目实地调研结果表明，航道工程空间分

2.2.2 加载矢量数据

1) 将关键数据(SHP)利用GIS软件转化为带有地理信息Geojson的数据格式;

2) 在 Unity 3D 中读取 Geojson 格式数据，并进行数据渲染。

3 三维航道数据组织与 LOD 技术

3.1 项目概况

长江干线武汉—安庆段上起天兴洲长江大桥、下迄安庆皖河口，全长约 386.5 km，工程主要针对湖广—罗湖洲、沙洲、戴家洲、鲤鱼山、张家洲、马当东流等 7 个碍航段进行系统整治，其中建设护滩带 28 道、坝体 5 道、高滩守护 15.4 km、护岸加固 13.3 km、基建疏浚 477 万 m^3 及生态建设工程(图 4)。该工程的 BIM 模型、高程、影像、矢量数据等 GIS 数据体量非常大，对三维数据在 Unity 引擎中加载与渲染提出巨大挑战。

3.2 航道工程分布特点

布特征主要为跨度大、点状分布、多点线性，总体沿长江沿岸进行相关工程建设(图5)。



图 5 长江武汉段某区域整治工程平面位置分布

3.3 三维航道数据组织管理

对于航道工程大场景数据而言,任何计算机一次性加载和渲染数据性能都会出现卡顿,因此要尽可能采用分区域数据组织调度技术。将一个区域分解成多个区域,一般是根据实际场景大小灵活分区,图 6 为某航道工程分区。

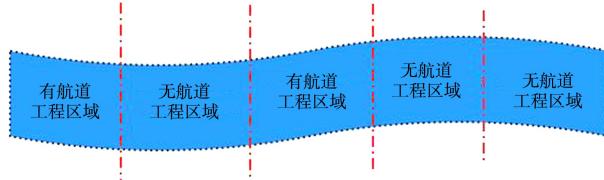


图 6 航道工程平面分区

将整个航道区域数据按照有、无航道工程区域以及单个区域大小分割成若干小区域,依据“动态导入、动态调用、动态卸载”原则将进入视野区域的数据导入,视野外区域数据则不导入,当已经导入区域数据在视野之外时,则对数据进行卸载,保证计算机实时加载数据量上限,达到节省资源的目的。

3.4 三维航道数据 LOD 技术

对于复杂数据场景来说,多的不仅是地形数据和 BIM 模型数据,场景中的文字、树、船等小品分布也较多,因此仅采用区域分块调度技术还无法达到最好的实时状态。因此对已加载模型数据通过 LOD(细节层次,levels of detail)技术进行渲染。其根据模型数据离视点距离来动态判定模型渲染的精细度,对于近处物体展示细节,对于远处物体只展示其外轮廓,从而节省计算机资源,效果如图 7 所示。

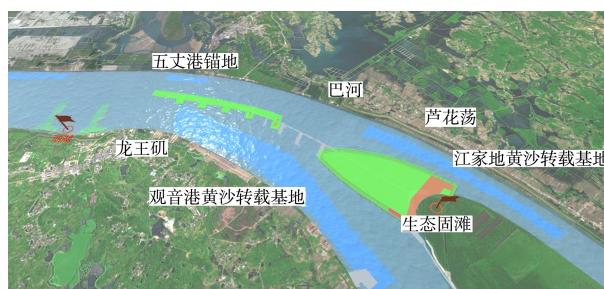


图 7 某航道区域三维场景

4 三维航道漫游系统实现

4.1 手动控制漫游

手动控制漫游是使用者通过手动操作键盘和

鼠标来浏览三维场景,实现对三维场景的缩放、旋转、平移等功能,灵活控制场景的视角^[4]。手动控制漫游一旦接受到相应的控制命令就立即响应,相机空间位置改变,浏览到相应场景,整个过程中场景不发生交互,只移动相机方位。

手动控制漫游实现步骤如下:

- 1)开启手动漫游脚本后,首先设置好控制方式:键盘控制,鼠标控制,同时控制,是否开启旋转控制、移动控制等;
- 2)接收用户输入的控制信息,传递到函数 Get Input Translation Direction 中(图 8),返回对应的移动方向;
- 3)得到移动方向后,计算出移动距离,移动相机到相应位置。

```
Vector3 GetInputTranslationDirection()
{
    Vector3 direction = new Vector3();
    if (UseKeyboard)
    {
        if (Input.GetKey(KeyCode.W))
        {
            direction += Vector3.forward;
        }
        if (Input.GetKey(KeyCode.S))
        {
            direction += Vector3.back;
        }
        if (Input.GetKey(KeyCode.A))
        {
            direction += Vector3.left;
        }
        if (Input.GetKey(KeyCode.D))
        {
            direction += Vector3.right;
        }
        if (Input.GetKey(KeyCode.Q))
        {
            direction += Vector3.down;
        }
        if (Input.GetKey(KeyCode.E))
        {
            direction += Vector3.up;
        }
    }

    if (UseMouse)
    {
        if (Input.GetMouseButton((int)TranslateBtn))
        {
            //Mouse pointer.
            var mouseX = Input.GetAxis("Mouse X");
            var mouseY = Input.GetAxis("Mouse Y");

            direction -= Vector3.right * mouseX;
            direction -= Vector3.up * mouseY;
        }
        direction += Vector3.forward * Input.mouseScrollDelta.y;
    }
    return direction;
}
```

图 8 自动漫游控制函数

4.2 自动漫游

场景的自动漫游不需要用户进行控制。其设计思想:首先确定漫游视点,每个漫游视点对应一个漫游时间,表示相机漫游到该视点所需要的时间,第 1 个视点对应的漫游时间为 0,最后 1 个视点对应的漫游时间为整段漫游总时间。将视点依次连接形成一条完整的路径,并根据采集视点

进行内插, 得到一条完整平滑连续的路径就是相机的漫游路径, 当开启自动漫游功能后, 相机沿着该路径在规定时间完成自动漫游^[5]。

4.2.1 设置漫游路径

在自动漫游过程中路径的设置主要包括关键漫游点的设置和根据关键节点内插成漫游路径(图9)。用户在三维场景内选取关键漫游点, 并在该漫游点设置文本、图片、视频等内容状态, 通过内插方式获得整条路径的三维点集, 为自动漫游提供数据。

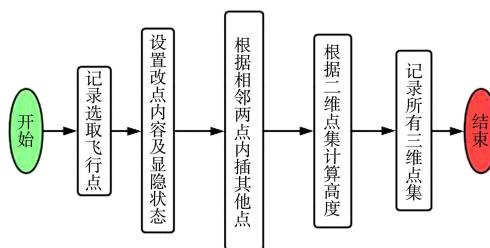


图9 路径设置步骤

飞行路径的插值计算步骤如下:

1) 获取路径相邻两个关键点平面坐标;

2) 根据相邻两点的距离和平滑指数, 内插出两点间其他二维平面坐标;

3) 从起始点至终点循环该过程, 得到整条路径所有二维平面点集;

4) 根据已有相邻点高程, 采用距离比例法计算高度值 H 。

一般插值方法有折线法和样条曲线法。由于折线法容易出现场景跳动, 漫游效果较差, 所以本文采用样条曲线插值法, 用样条函数计算插值点。

4.2.2 自动漫游飞行

自动飞行主要通过定时器来实现。漫游开始后定时器根据已经设定好的路径点集在设置时间执行相应指令, 并绘制渲染当前相机位置的姿态。当读取到的漫游点数据包含数据内容时, 定时器会自动获取所设置的内容, 并根据内容状态设置显示和隐藏, 从而达到动态内容效果, 保证整个漫游过程不间断、不跳帧、不跳跃连续的自动飞行漫游效果(图10)。



图10 自动漫游效果

5 结语

1) 根据航道工程跨度大、点状分布、多点线性空间分布特性, 提出分区数据组织调度技术, 将广域航道工程分成若干区域, 区域之间依据“动态加载、动态渲染、动态卸载”原则进行动态调度, 降低计算机实时处理数据量, 解决航道工程大体量数据加载问题。

2) 对单一区域模型数据通过 MeshSimplify 插

件对模型进行轻量化处理, 减少数据的体量, 对比传统人工手动减面大大较少工作量、节省工作时间; 在 Unity 3D 实时渲染过程中采用 LOD 技术动态渲染数据, 利用 Geojson 格式数据对 SHP 数据进行加载与解析, 解决 Unity 3D 加载大体量 BIM+GIS 数据渲染不流畅、卡顿问题。

(下转第 178 页)